Representation and Optimal Recognition of Human Activities

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

Abstract

Towards the goal of realizing a generic automatic human activity recognition system, a new formalism is proposed. Activities are described by a chained hierarchical representation using three type of entities: image features, mobile object properties and scenarios. Taking image features of tracked moving regions from an image sequence as input, mobile object properties are first computed by specific methods while noise is suppressed by statistical methods. Scenarios are recognized from mobile object properties based on Bayesian analysis. A sequential occurrence several scenarios are recognized by an algorithm using a probabilistic finite-state automaton (a variant of structured HMM). The demonstration of the optimality of these recognition method is discussed. Finally, the validity and the effectiveness of our approach is demonstrated on both real-world and perturbed data.

1 Introduction

Automatic or semi-automatic recognition of human activity is gaining attention in computer vision research community due to the needs of many applications such as surveillance for security and for human-computer interaction. Automatic human activity recognition by computer involves, firstly, detecting and tracking mobile objects from the image sequence captured from the domain of interest. The activities are then recognized from the characteristics of these tracked mobile objects. One of the major tasks in this process is concerned with how to link the gap between numerical pixel level data and a high level abstract activity (verbal) description. This leads to a spectrum of approaches, which interprets this task as a matching process between a sequence of image features to a set of activity models. The best matched models are then selected based on some criteria and their matching degree. The differences among these approaches are 1) whether image features are computed automatically and independently of input image sequences, 2) whether the activity representation is generic and expressive enough to model a variety of activities but yet powerful enough to discriminate between similar activities (e.g. *sitting* and *squatting*), and 3) whether the matching is performed optimally (if so, what are the underlined optimality criteria and assumptions).

In a previous paper [5, 4], we introduced an activity recognition framework that allows a flexible and extendible representation of activities using a hierarchical model and demonstrated it on different applications such as monitoring pedestrians at a road block and cars on a freeway. In this framework, image features are linked explicitly to a symbolic notion of activity through several layers of more abstract activity descriptions combined by a variety of methods. The goal of this paper is to present a recognition algorithm that effectively recognizes image sequences that correspond to a particular scenario model.

This paper is organized as follows. Related work is discussed in section 2. Section 3 describes our formalism concerned with the activity modeling and recognition methods. An algorithm for optimally recognizing scenarios based on Bayesian analysis and finite-state automaton are described in detail in section 4 together with the optimality criteria. Experimental results are presented in section 5.

2 Related Work

Activity recognition systems that have been developed in the recent past mostly provide a representation and recognition techniques for one particular type of action in a particular domain and sometimes under a constrained environment. Many of these methods are targeted at *simple events* defined as events that can be detected by itself (i.e. context-free) regardless of other events. For example, *"sitting"* or *"walking"* can be detected independently while *complex events* such as *"approaching another person, handing an object, then walking away"*, are composed of three sub-events with a constraint on the sequential occurrence of them (hence, is context-sensitive).

In [7], simple periodic events (e.g. walking) are recognized by constructing dynamic models of periodic pattern of people's movements and is dependent on the robustness of tracking. Inspired by a similar application to speech recognition, *Hidden Markov Model*(HMM) has also been

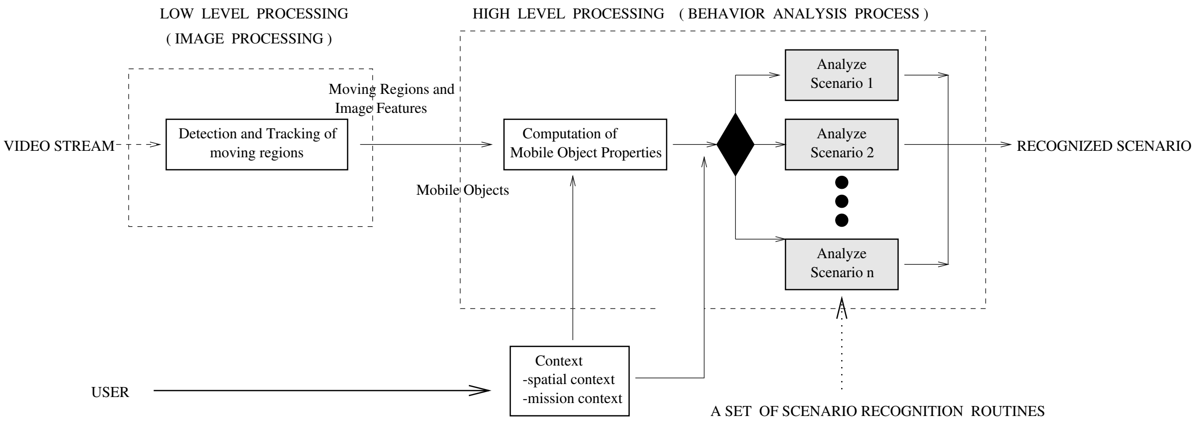


Figure 1: *Overview of the system*

applied to activity recognition. In [10], an HMM is used as a representation of simple events which are recognized by computing the probability that the model produces the visual observation sequence. Parameterized-HMM [11] and coupled-HMM [3] are introduced to recognize more complex events such as an interaction of two mobile objects. In [2], stochastic context-free grammar parsing algorithm is used to compute the probability of a temporally consistent sequence of primitive actions recognized by HMMs. Even though HMMs are robust against various temporal segmentations of events, the structure and probability distributions are not transparent to human and need to be learned using iterative methods. For complex events (e.g. a combination of sub-events) such networks and their parameter space may become prohibitively large.

Bayesian networks have been adopted in activity recognition in the recent years mostly to represent simple events [6, 1, 8]. To effectively apply Bayesian networks in activity recognition, we need the correct structure and conditional probabilities associated with the links in the network. In [9], a Bayesian network is used to recognize the action *"sitting"* by making an assumption that this action is related with only the change of location of human head (which is manually tracked) across time. This assumption simplifies the network structure and parameters but the network may fail to discriminate between *"sitting"* and other similar actions (e.g. *"bending over"*) in a real application. In [8], a more complicated Bayesian network is defined together with specific functions to evaluate some temporal relationship among events (e.g. *before* and *around*) to recognize actions involving multiple agents in a football match. However, the tracking of football players is performed manually.

3 A Formalism for Human Activity Recognition

Figure 1 shows schematically how mobile objects and their behavior may be recognized among a large number of potential scenarios from input visual data (image sequence) and available context. **Context** consists of associated information, other than the sensed data, that is useful for activity recognition such as a spatial map and prior activity expectation. A hierarchy of entities, which consist of three levels: **image feature**, **mobile object property**, and **scenario** is defined. Figure 2 shows an example of activity representation based on this hierarchy. At the lowest layer, image features are the input to the system. Several layers of mobile object properties and scenarios are then defined to describe a more complex and abstract activity shown at the highest layer.

From input image sequence, moving regions are detected and tracked, and several 2D (and, if available, 3D) **image features** are computed for these regions. Image features correspond to uninterpreted instantaneous data of a moving region (e.g. shape and location measurements) computed by some lower level image processing routines. These values are used as primitive input to the higher mobile object property level.

At this level, mobile objects are hypothesized as being composed of one or more tracked moving regions and *several layers* of spatial and temporal **mobile object properties** are computed over a few frames. Some of them can be generic and elementary such as width, height, color histogram or texture while the others can be complex (e.g. a graph description for the shape of an object). Properties can also be defined with regard to the context (e.g. some short events such as *"entering the security area"*).

The links between a mobile object property at a higher layer to a set of properties at the lower layers represent some relations between them (e.g. taking a ratio of width and height properties to compute the aspect ratio of the shape of mobile objects). A filtering function and a mean function that compute a mean value based on the multi-Gaussian distribution of the property values collected over time are also available to minimize the errors caused by environmental

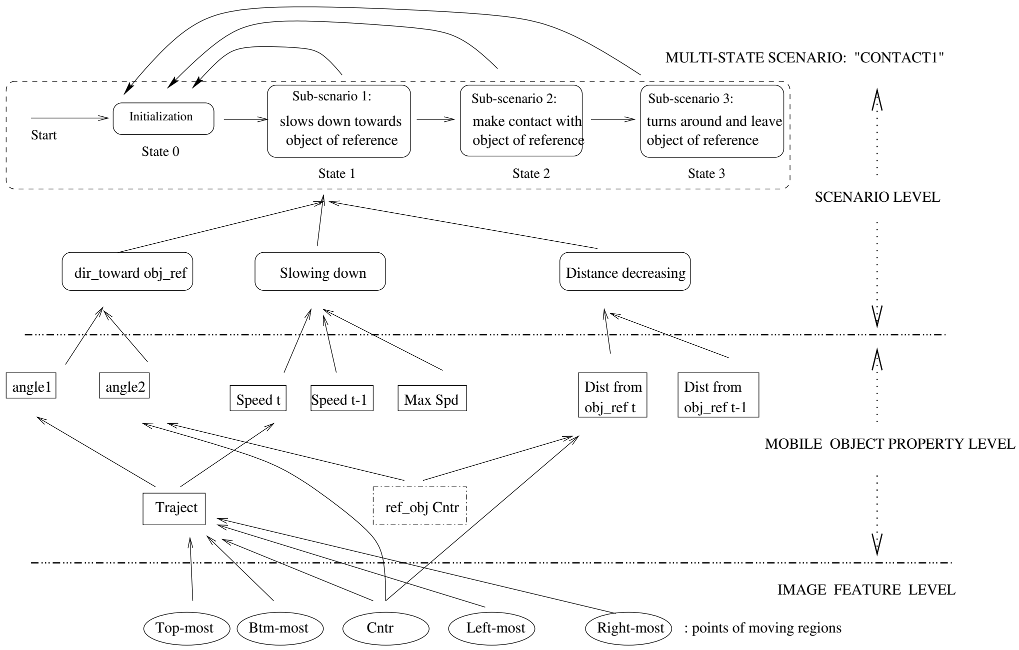


Figure 2: *Representation of activity "contact1" defined as a person walks towards another person, makes contact with, turns around and walks away. Image features are shown in oval, mobile object properties in rectangular, scenarios in rectangular with round corners, and context in dotted rectangular.*

and sensor noise.

Scenarios correspond to long-term activities of mobile objects such as *"a person walks towards another person, hands in an object, then walks away"*. Two types of scenarios (i.e. *single-state scenario* and *multi-state scenario*) are defined based on the type of the temporal combination of their composite sub-scenarios at lower levels.

Single-state scenarios are defined by a concurrent *logical constraint* on a set of sub-scenarios or mobile object properties. This constraint is verified at every frame to determine whether or not the single-state scenario has occurred at that instance. For example, scenario *"mobile object* **A** *is walking toward mobile object* **B***"* represents a logical constraint of two mobile object properties: *"the distance between* **A** *and* **B***"* and *"the direction of* **A***"*. If the distance decreases and the direction of **A** is toward **B**, then the scenario is said to be recognized.

Multi-state scenarios correspond to a temporal sequence of sub-scenarios (or a sequence of states, where states refer to sub-scenarios) and is verified over a long sequence of frames. For example, in figure 2, scenario *"contact1"* explained as *"mobile object* **A** *makes contact with mobile object* **B** *and rushes away"* represents a sequential occurrence of three sub-scenarios: *"***A** *slows down towards* **B***"*, then *"***A** *is in contact with* **B***"* and then *"***A** *turns around and goes away from* **B***"*. Multi-state scenarios are represented by a finite-state automaton where the states of the automaton correspond to the composite sub-scenarios.

Scenarios can be organized into a library and triggered as needed. Scenarios have a confidence value (or a probability distribution) attached to them based on statistical analysis. In this representation, the links from image features to high level activities are explicit and can be constructed naturally by users.

4 Optimal Activity Recognition

Given a set of scenario models defined by our proposed representation (S = $scenario_1$, $scenario_2$,..., $scenario_N$) and an observation sequence of mobile object properties ($O = O_{(1,t)} = O_1 O_2 ... O_t$, where O_i is composed of a set of mobile object properties), we wish to compute the likelihood of the scenarios that have occurred. The optimal criteria here is to find the scenario model that is most likely to produce the sequence of observations. We consider a scenario as having a binary distribution (i.e. whether it occurs or does not occur). We want to compute $\forall i, P(scenario_i | O_{(1,t)})$, and find the one with the maximal value.

$P(scenario|O_{(1,t)})$, can be computed by inferring the distribution of sub-scenario values at the lower layers and propagating them towards the top layer. We describe in this section the methods that we use to infer these values.

4.1 Single-state Scenario Recognition

To infer whether or not a single-state scenario occurs at any frame from the observed distributions of sub-scenarios, we apply Bayesian methods. In Bayesian terms, the input entities are viewed as providing the evidence variables and the task is to compute the probability distribution of an output entity hypothesis. Bayesian methods are optimal provided that the joint probability distributions needed to carry out the computations are known. If the entities to be combined are statistically independent (given the scenario), a simple *naive* Bayes classifier can be used to compute the distribution of the combined result. When the entities are not conditionally independent, Bayesian networks offer an efficient representation that can represent dependence by decomposing the network into conditionally independent components.

We apply Bayesian networks in our formalism as follows. The structure of the network and many probabilities are derived by heuristic means from the knowledge about the domain. For example, in our hierarchical representation, logical constraints of sub-scenarios that represent the recognition of a particular scenario are a strong indication of causality between them. Each layer in the hierarchy can be viewed as being composed of several naive Bayesian classifiers (one classifier per each scenario). That is, each scenario in the layer becomes a parent node of a naive Bayesian classifier which links to other sub-scenarios (child nodes) at lower layers. Belief propagation (inference process) is performed in one direction from the bottom layer to the top layer.

Two layers of naive Bayesian classifiers are defined for the representation in figure 2. At the top layer, as a sub-scenario 1 of the automaton, *"slowing down toward object of reference"*, (called parent scenario) from three other sub-scenarios: *"direction is toward the object of reference"* (e_1), *"slowing down"*(e_2), and *"distance is decreasing"* (e_3) (called child scenarios). These child scenarios form another layer of three naive Bayesian classifiers (e.g. e_1 becomes a parent of *"$angle_1$"* and *"$angle_2$"*). The distribution over the parent scenario values in Bayesian classifiers are inferred from the distribution of sub-scenario values and the conditional probabilities of sub-scenarios given the values of parent scenario (i.e. $P(e_1|H)$, $P(e_2|H)$, and $P(e_3|H)$). In some cases, these probabilities can be determined heuristically (e.g. strictly logical functions such as H is true if e_1, e_2, and e_3 are true). In the case where these cannot be determined heuristically, they can be learned from observed examples by simply making a histogram of observed values of the evidence variables, e_1, e_2, e_3, given the value of a given hypothesis, H, i.e. to compute the probability distribution $P(e_i|H)$ and $P(e_i|\neg H)$. By taking advantage of the fact that the nodes of the network are transparent (e.g. we can observe whether the object is moving towards another object or whether it is slowing down), this parameter learning process becomes simple.

4.2 Multi-state Scenario Recognition

The structure of an automaton (i.e. states and transitions) that represents a multi-state scenario can be easily obtained by forming a sequence of DAGs of composite sub-scenarios $(S_1, S_2, ..., S_N)$ where S_i refers to sub-scenario i (or state i). The inter -temporal relationship between sub -scenarios is modeled by a transition. In our case, state i either advances to state $i+1$, remains in the same state, or goes back to the initial state. A multi-state scenario is said to occur when its sub-scenarios are recognized consecutively. Given the observation sequence, the question we ask is when is the most likely transition time from one state to the next so that the whole state sequence is completed (i.e. we are interested in the transition timing that maximizes the probability of the sequence having occurred). We define the probability of such an event as $P(MS^*|O)$: *the probability that the complete automaton state sequence of MS occurs with the most likely state transition timing given the sequence of observations $O = O_{(1,t)}$*. This can be computed as follows:

$$P(MS^*|O) = \max_{\forall(t_1,t_2,...,t_N)} P(S_{1_{(t_1,t_2-1)}} S_{2_{(t_2,t_3-1)}} ... S_{N_{(t_N,t)}}|O), \quad (1)$$

where t_i refers to the time that the transition to state i from state $i-1$ occurs and $S_{i_{(t_i,t_{i+1}-1)}}$ means that scenario i occurs during t_i and t_{i+1} - 1.

To be concise, let S_1^N be $S_{1_{(t_1,t_2-1)}} S_{2_{(t_2,t_3-1)}} ... S_{N_{(t_N,t)}}$. The computation of $P(S_1^N|O)$ can then be decomposed into the recognition results of sub-scenarios as:

$$P(S_1^N|O_{(t_1,t)}) = \prod_{1\leq i\leq N} \frac{a_{i,i-1} P(S_{i_{(t_i,t_{i+1}-1)}}|O_{(t_i,t_{i+1}-1)})}{P(S_{i_{(t_i,t_{i+1}-1)}})}. \quad (2)$$

where $t_{N+1} = t$, and $a_{i,i-1}$ is the *a priori* probability of the transition from state $i-1$ to state i at time t_i (i.e. $P(S_{i_{(t_i,t_{i+1}-1)}}|S_{i-1_{t_i-1}})$) which is assumed to be constant for all scenarios. $\prod_{1\leq i\leq N} P(S_{i_{(t_i,t_{i+1}-1)}})$ is a characteristic of the scenario and can be learned from image sequences or provided as context. In the case where we consider similar or competing scenarios, we can make a fair assumption that this product is a constant.

Eq. 2 can be derived as follows. For compactness, we drop the timing notation symbols in the proof. Let S_i be $S_{i_{(t_i,t_{i+1}-1)}}$ and O_i be $O_{(t_i,t_{i+1}-1)}$. We have that,

$$P(S_1^N|O) = P(S_N, S_1^{N-1}|O_N, O_1^{N-1}) \quad (3)$$

$$= P(S_1^{N-1}|S_N, O_1^{N-1}, O_N) P(S_N|O_1^{N-1}, O_N) \quad (4)$$

$$= P(S_1^{N-1}|S_N, O_1^{N-1}) P(S_N|O_N) \quad (5)$$

$$= \frac{P(S_N|S_1^{N-1}, O_1^{N-1}) P(S_1^{N-1}|O_1^{N-1}) P(S_N|O_N)}{P(S_N|O_1^{N-1})} \quad (6)$$

$$= \frac{P(S_N|O_N) P(S_N|S_1^{N-1}) P(S_1^{N-1}|O_1^{N-1})}{P(S_N)} \quad (7)$$

Eq. 4 and 6 is achieved from the conditional probability axiom, while eq. 5 is achieved from the fact that given O_N, S_N is independent of O_1^{N-1} and vice versa. Eq. 7 is achieved from the assumption that S_N and O_1^{N-1} are independent of each other. By making a Markov assumption that the probability of staying in state i at time t only depends on the likelihood of being in state $i-1$ at time $t-1$, we have that

$$P(S_N|\, S_1^{N-1}) = P(S_{N_{t_N}}|S_{N-1_{t_{N-1}}}) = a_{N,N-1}. \quad (8)$$

We can recursively apply eq. 7 to $P(S_1^{N-1}\mid O_1^{N-1})$ and finally get eq. 2. By substitute eq. 2 into eq. 1, we have:

$$P(MS^*|O) = \max_{\forall(t_1,\dots,t_N)} \prod_{1\le i\le N} \frac{a_{i,i-1}P(S_{i_{(t_i,t_{i+1}-1)}}|O_{(t_i,t_{i+1}-1)})}{P(S_{i_{(t_i,t_{t+1}-1)}})} \quad (9)$$

We describe next the algorithm to find the values of $t_1, t_2, ..., t_N$ that maximize $P(MS^*|O)$.

Multi-state Scenario Recognition Algorithm

The direct computation of $P(MS^*|O)$ at time T using eq. 9 involves an operation of $O(T^N)$ complexity since there are $O(T^N)$ combination of values of $t_1, t_2, ..., t_N$. However a more efficient recursive algorithm based on dynamic programming can be applied. This algorithm is an adaptation of the Viterbi algorithm used in HMM to our finite-state automaton. Let $R_i(t)$ be the likelihood that MS occupies state i at time t with the most likely transition timing between states given the observation sequence $O_{(t_1,t)}$. That is,

$$\begin{aligned} R_i(t) &= \max_{\forall(t_1,t_2,\dots,t_i)} P(S_{1_{(t_1,t_2-1)}}S_{2_{(t_2,t_3-1)}}...S_{i_{(t_i,t)}}|O_{(t_1,t)}) \\ &= \max_{\forall(t_1,t_2,\dots,t_i)} \prod_{1\le j\le i} \frac{a_{j,j-1}P(S_{j_{(t_j,t_{j+1}-1)}}|O_{(t_j,t_{j+1}-1)})}{P(S_{j_{(t_j,t_{j+1}-1)}})} \end{aligned} \quad (10)$$

The solution for eq. 1 is, therefore, equivalent to $R_N(t)$. $R_i(t)$ can be derived from previously recognized state $i-1$ as follows. For short, let A_j be the term on the right hand side of the product.

$$\begin{aligned} R_i(t) &= \max_{\forall(t_1,t_2,\dots,t_i)} \prod_{1\le j\le i} A_j \\ &= \max_{t_{i-1}\le t_i\le t} (A_i \max_{\forall(t_1,t_2,\dots,t_{i-1})} \prod_{1\le j\le i-1} A_j) \\ &= \max_{t_{i-1}\le t_i\le t} A_i R_{i-1}(t_i-1) \end{aligned} \quad (11)$$

By substitute A_i back in eq. 11, we have that,

$$R_i(t) = \max_{t_{i-1}\le t_i\le t} \frac{a_{i,i-1}P(S_{i_{(t_i,t)}}|O(t_i,t))}{P(S_{i_{(t_i,t)}})} R_{i-1}(t_i-1) \quad (12)$$

$$t_{i_{best}} = \underset{t_{i-1}\le t_i\le t}{argmax} \frac{a_{i,i-1}P(S_{i_{(t_i,t)}}|O_{(t_i,t)})}{P(S_{i_{(t_i,t)}})} R_{i-1}(t_i-1) \quad (13)$$

At time t, starting from state 1 where $R_0(t)$ is always 1, eq. 12 are recursively processed until the final state N is reached, where $R_N(t)$ represents the probability of the sequence of states occurs with optimal transition timing $t_{1_{best}}, t_{2_{best}}, ..., t_{N_{best}}$.

In eq. 12, $P(S_{i_{(t_i,t)}}|O_{(t_i,t)})$ is the probability that sub - scenario i (which can be either a single - state or a multi - state scenario) occurs given that we observe the observation sequence from time t_i to $t_{i+1}-1$. This can be computed as a temporal mean (expected value) of the distribution of $P(S_{i_t}|O_t)$ collected from frame t_i to t as described below.

Temporal Mean Computation of $P(S_{i_{(t_i,t)}}|O_{(t_i,t)})$

The distribution of scenario values during any time period, assuming that the noise is random, should distribute around either zero or one (i.e. occurring or not occurring). For each candidate value of t_i, we collect scenario values (i.e. *the probability* that it occurs) during t_i and t and compute *the expected scenario value*. The expected scenario value represents the temporal mean of the distribution of $P(S_{i_{(t_i,t)}}|O_{(t_i,t)})$.

To find the optimal t_i that maximizes $R_i(t)$, we would have to investigate all possible values of each t_i. However, there is a criteria to disregard many of these values as a candidate for $t_{i_{best}}$. Let t'_i be a possible value of t_i. One indicator for t'_i to be disregarded as a potential candidate for t_i is the fact that the accumulative probability that scenario i does not occur during t'_i and t is greater than the accumulative probability that scenario i does occur, which indicates that scenario i does not occur during t'_i and t. In general, only a certain numbers, say k, of such t_i candidates ($t_i^k = t_{i_1}, ..., t_{i_k}$) that compute the highest $R_i(t)$ can be maintained.

The algorithm for the computation of temporal mean is summarized as follows. Let 1) $S_i^+(t'_i,t)$ be the accumulative probability density of the scenario (above threshold δ which is inversely proportionate to the degree of noise in the image sequences) from time t'_i to t, 2) $S_i^-(t'_i,t)$ be the accumulative probability density when the scenario is not recognized (i.e. under the threshold δ) from time t'_i to t, and 3) $E[S_i]_{(t'_i,t)}$ be the expected recognition value during t'_i and t.

- *$S_i^+(t'_i,t)$ and $S_i^-(t'_i,t)$ computation:*

$$S_i^+(t'_i,t) = \sum_{t'_i\le j\le t, P(S_{i_j})>\delta} P(S_{i_j}),$$

$$S_i^-(t'_i,t) = \sum_{t'_i\le j\le t, P(S_{i_j})\le\delta} (1-P(S_{i_j}))$$

- *$E[S_i]_{(t'_i,t)}$ computation:*
  If $(S_i^+(t'_i,t) > S_i^-(t'_i,t))$,

then $P(S_{i_{(t'_i,t)}}|O_{(t'_i,t)}) = \frac{S_i^+(t)}{t-t'_i}$
else,$P(S_{i_{(t'_i,t)}}|O_{(t'_i,t)}) = 0$ and t'_i is disregarded.

In term of complexity, if only a certain number k of t_i are maintained, we need to update k number of temporal means to compute $R_i(t)$. Since this process is repeated for all N states, multi-state scenario recognition algorithm requires $O(NT)$ operations, which is as efficient as the Viterbi algorithm used in the HMM approach. However, the construction of the model, and the initialization and learning of the parameters can be much easier in our case since the nodes of the network are transparent.

5 Results

We validate the proposed formalism by analyzing ten video streams that capture all activities in two different parking lot areas. The parameters of the network are learned from sequences taken at the first parking lot. The training data set is composed of 600 frames (containing half of positive and half of negative examples). We then test the network on the sequences taken from the both parking lots; the second lot has a different street structure (the streets are of high curvature). We show results of two test sequences called sequence **A** and **B** shown in figure 3 (a) and 5 (a) taken in the second parking lot.

Activity Description

We model two activities to be detected using our hierarchical representation. First activity, *"Contact1"* is defined as a multi-state combination (automaton) of three sub-scenarios: *"two men approach each other, make contact, turn around and leave"*. These sub-scenarios are then described by the network shown in figure 2. The second activity, *"Passing By"* is defined as *"A man approaches another man, walks past the other, and then leaves"*, and is described by a network similar to that for "Contact1", but with different sub-scenarios.

Activity Recognition

The analysis results for sequence **A** are shown in figure 3 (b). Dotted, dashdot and solid lines show the likelihood of the sub-scenarios which are derived using Bayesian networks described in section 4.1. The "+" line shows $P(MS^*|O)$ computed based on our multi-state scenario recognition algorithm. The results show that "Contact1" is recognized with higher confidence ($P(MS^*|O)$ = 0.7) compared to "Passing By" ($P(MS^*|O)$ = 0).

To validate the robustness of our method against a variety of temporal segmentation of activities and levels of noise, we performed an experiment on four hundred perturbed sequences. Perturbed sequences were generated from sequence **A** by randomly inserting and deleting the tracked mobile objects as follows. A random number uniformly distributed between 1 and 20 is generated and used as a timing point where the sequence is perturbed. For each perturbed sequence, three such random numbers are selected and used as timing point to duplicate or delete a frame or to remove the information on the tracked blobs. For example, if the random number is 3, a frame may be inserted once in every three frame and so on. The smaller the random number is the more the sequence becomes perturbed. As a result, mobile objects may appear to be static or lost from time to time. The probabilities of scenario "Contact1" at the final frame of each processing are shown in figure 4 (a). It shows that on average "Contact1" is recognized with a probability of 0.55 (with the highest value of 0.74 and lowest value of 0.21). Figure 4 (b) shows the result of recognition on a sequence where the tracked moving blobs were removed once in every three frames. From the characteristics of the "+" line (i.e. very stable), our method still infers the correct scenario.

Figure 5 (b) shows the analysis results for sequence **B**. This sequence depicts scenario "Passing By" which is correctly recognized with probability of 0.6, while the scenario "Contact1" is poorly recognized at the lower value of 0.2.

6 Conclusion

We presented a new formalism to recognize various kinds of activities and differentiate between similar ones. This formalism contains a chained hierarchical representation that describes scenarios from general properties of the moving objects. We claim that our proposed formalism is optimal based on two reasons. First, the formalism allows us to take advantage of any mobile object properties which are obtained from the statistical processing of the output of image processing routines. Second, these mobile object properties can be combined in a probabilistic framework (Bayesian networks and probabilistic automaton) such that the scenario with the maximum probability of occurrence can be selected. Our experiments indicate the validity of our approach. We plan to conduct more extensive tests to establish the generality of our approach.

Acknowledgements

This work was in part supported by the Defense Advanced Research Projects Agency of the U.S. Government under contract DACA 76-97-K-0001.

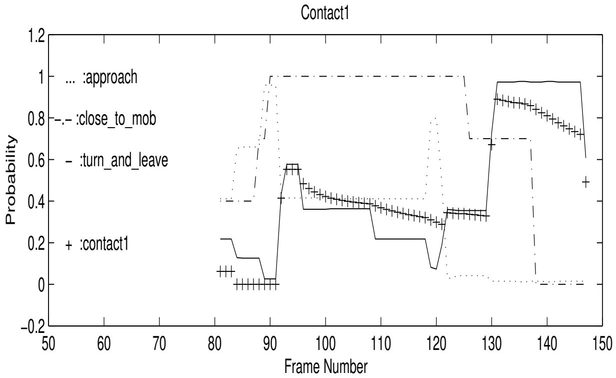
References

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

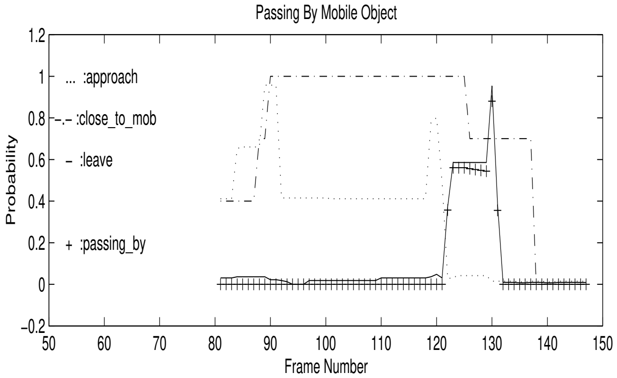
[2] A. Bobick and Y. A. Ivanov. Action recognition using probabilistic parsing. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998.

frame A.63 frame A.95 frame A.111 frame A.157

(a) Detection and tracking of moving regions for scenario "CONTACT1".



I) CONTACT1



II) PASSING_BY

(b) Recognition results of two competing activities.

Figure 3: *(a) Input sequence* **A** *shows a multi-state scenario "Contact1". Object 1 (at the top) approaches object 2 (at the bottom), makes contact (both objects have merged as they meet), turns around and leaves. (b) Scenario "Contact1" is recognized with $P(MS^*|O)$ = 0.7. On the other hand, scenario "Passing By" is recognized with lower probability (almost 0 at the end) since sub-scenario "leaving without turning around" is not established.*

[3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, Puerto Rico, USA, 1997.

[4] F. Brémond and G. Medioni. Scenario recofnition in airborne video imagery. In *DARPA Image Understanding Workshop*, pages 211–216, 1998.

[5] F. Brémond and G. Medioni. Scenario recognition in airborne video imagery. In *the Workshop of Computer Vision and Pattern Recognition on Interpretation of Visual Motion*, Santa Barbara (California), June 1998.

[6] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1-2):431–459, 1995.

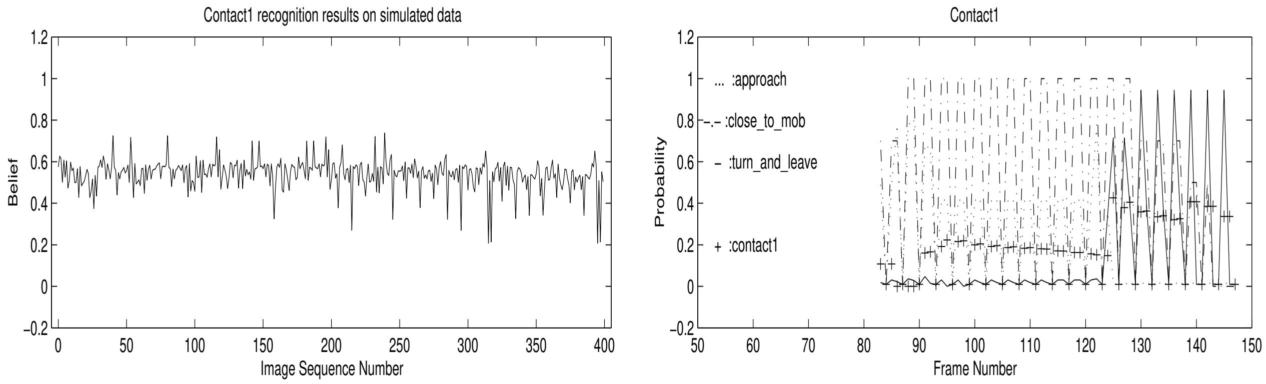
[7] L. Davis, R. Chelappa, A. Rosenfeld, D. Harwood, I. Haritaoglu, and R. Cutler. Visual surveillance and monitoring. In *DARPA Image Understanding Workshop*, pages 73–76, 1998.

[8] S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *IEEE Proceedings of the National Conference on Artificial Intelligence*, April 1999.

[9] A. Madabhushi and J. K. Aggarwal. A bayesian approach to human activity recognition. In *Second IEEE International Workshop on Visual Surveillance (CVPR Workshop)*, pages 25–30, Fort Collins, CO, June 1999.

[10] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of ISCV '95*, 1995.

[11] A. Wilson and A. Bobick. Recognition and interpretation of parametric gesture. In *IEEE Proceedings of the International Conference on Computer Vision*, Bombay, India, 1998.



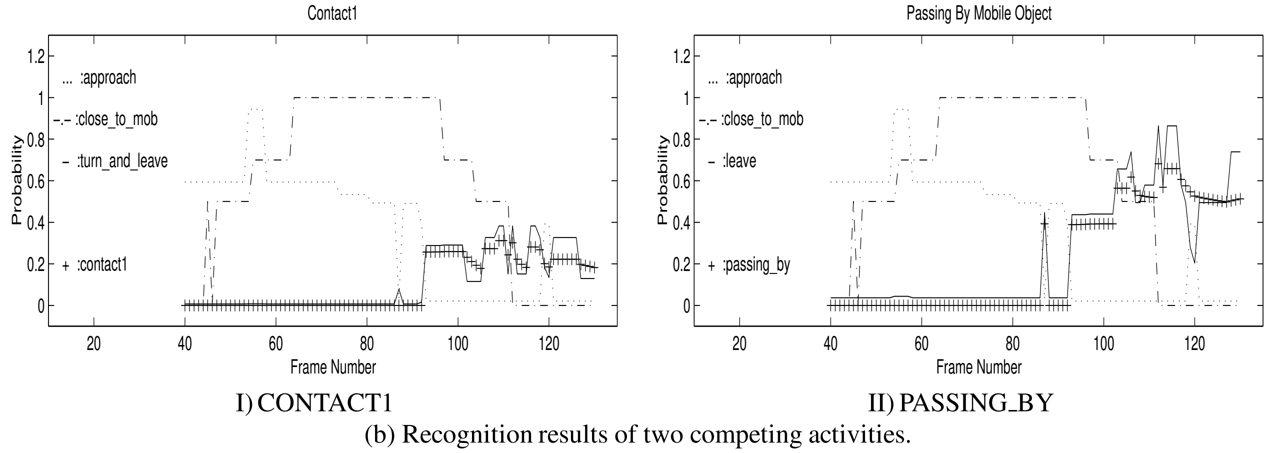
(a) Results on perturbed sequences

(b) One example of perturbed data sequence

Figure 4: *(a) "Contact1" recognition values of 400 perturbed image sequences. (b) The recognition result on an example of perturbed sequence with the loss of tracking on every three frames. The recognition of sub-scenarios becomes zero every time the mobile object is lost. Scenario "CONTACT1", however, is still correctly recognized.*

frame B.97 frame B.101 frame B.108 frame B.138

(a) Detection and tracking of moving regions for scenario "PASSING-BY"



I) CONTACT1

II) PASSING_BY

(b) Recognition results of two competing activities.

Figure 5: *(a) Input sequence* **B** *shows a multi-state scenario "Passing By":Object 1 walks past object 2. (b) "Passing By" is recognized with higher confidence ($P(MS^*|O)$ = 0.6) than "Contact1"($P(MS^*|O)$ = 0.2).*