International Journal of Pattern Recognition and Artificial Intelligence
© World Scientific Publishing Company

RECOGNIZING OVERLAPPED HUMAN ACTIVITIES FROM A SEQUENCE OF PRIMITIVE ACTIONS VIA DELETED INTERPOLATION

KRIS M. KITANI
YOICHI SATO
Institute of Industrial Science,
The University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan
kitani@iis.u-tokyo.ac.jp
ysato@iis.u-tokyo.ac.jp

AKIHIRO SUGIMOTO
National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
sugimoto@nii.ac.jp

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

The high-level recognition of human activity requires a priori hierarchical domain knowledge as well as a means of reasoning based on that knowledge. Based on insights from perceptual psychology, the problem of human action recognition is approached on the understanding that activities are hierarchical, temporally constrained and at times temporally overlapped. A hierarchical Bayesian network (HBN) based on a stochastic context-free grammar (SCFG) is implemented to address the hierarchical nature of human activity recognition. Then it is shown how the HBN is applied to different substrings in a sequence of primitive action symbols via deleted interpolation (DI) to recognize temporally overlapped activities. Results from the analysis of action sequences based on video surveillance data show the validity of the approach.

Keywords: Activity Recognition; Context-Free Grammars; Bayesian Networks; Deleted Interpolation; Syntactic Pattern Recognition.

1. Introduction

The automated real-time understanding of human activities from a video sequence is a topic of growing interest in recent times. In the field of video surveillance, detecting suspicious activity in real-time would mean stopping crimes while they are happening or even before they happen. In application to human-computer interfaces, computers could adjust according to the activity context of the user. An intelligent system that recognizes high-level human activities offers a wide range of applications to aid people in everyday activities.

To implement a system that recognizes human activities, our task is two-fold. First, we need a psychological framework for characterizing human activity. Second, we need a computational method of analyzing those activities.

The characteristics of human activities can be learned from perceptual psychology[1]. Activities are hierarchical. They are organized, existing at various levels of abstraction. For example, walking and running are a *type of* moving. Activities are also partonomical meaning that primitive actions are temporally ordered (sequential). For example, the activity of *leaving an object* in a room might consist of a sequence of primitive actions: (1) enter the room, (2) put down the object and (3) exit the room. Activities can also be temporally overlapped. For example, the transition of a person *walking through* a room might overlap with the activity of the person *departing* from the room. From our perspective, it is difficult to identify the exact time at which the activity *walking through* has ceased and when the activity *departing* has started. Thus there is an inherent ambiguity at transitions between human activities which should be represented by a cognitive system.

To address the latter half of the problem, namely the computational recognition of human activities from a sequence of video images, we need an efficient method of incorporating the characteristics of activity mentioned above. The recognition system must encode hierarchical information, capture temporally constrained activities and accurately represent temporally overlapped activities.

Our contribution lies in the novel application of deleted interpolation (DI) – a smoothing technique used in natural language processing – for recognizing temporally overlapped activities. This paper addresses the issue of hierarchical structure by implementing a stochastic context-free grammar (SCFG). We convert the SCFG into a Bayesian network (BN) to create a hierarchical Bayesian network (HBN) which enables us to execute more complex probability queries across the grammar. We then apply the HBN via DI to a string of observed primitive action symbols to recognize various activities, especially those that are overlapped.

It is noted here that we are not directly addressing the issue of extracting symbols from a video sequence. Instead, we assume that a set of reliable low-level observations (e.g. appearance and movement attributes) are available to us, allowing us to focus on building up a scheme for activity recognition. Furthermore, the method of grammar creation is not the focus of this paper and therefore the grammar has been created heuristically.

2. Related Research

The majority of models that have been proposed for activity analysis are models that represent an activity as a sequential transition between a set of finite states (i.e. NDA[2], FSA[3], HMM[4], CHMM[5], VLHMM[6], LHMM[7], DMLHMM[8], ODHMM[9], SHSMM[10]). However, due to the fact that most simple activities do not have complex hierarchical structure, these models have not explicitly incorporated the concept of hierarchy into the model topology.

On the other hand, there have been a few works that have proposed hierarchical models to recognize structured activities. Contributions from computer vision started with Brand[11], when he utilized a deterministic action grammar to interpret a video sequence of a person opening a computer housing unit. Multiple parses over a stream of outputs from the low-level event detector were ranked and stored, giving priority to the highest ranking parse. Ivanov[12] first used a SCFG for action recognition using the Earley-Stolcke parser to analyze a video sequence of cars and pedestrians in a parking lot. Moore[13] also used a SCFG to recognize actions in a video sequence of people playing Blackjack. They extend the work of Ivanov by adding error correction, recovery schema and role analysis. Minnen[14] built on the modifications made by Moore by adding event parameters, state checks and internal states. They applied the SCFG to recognize and make predictions about actions seen in a video sequence of a person performing the Towers of Hanoi task. From a background in plan recognition, Bui[15] used a hierarchy of abstract policies using Abstract Hidden Markov Models (AHMM) implementing a probabilistic state-dependent grammar to recognize activities. The system recognizes people going to the library and using the printer across multiple rooms. AHMMs closely resemble the Hierarchical Hidden Markov Models (HHMM)[16] but with an addition of an extra state node. Nguyen[17] used an abstract Hidden Memory Markov Model (AHMEM), a modified version of the AHMM, for the same scenario as Bui.

The aforementioned works used domains with high-level activities delineated by clear starting points and clear ending points, where the observed low-level action primitives are assumed to describe a series of temporally constrained activities (with the exception of Ivanov[12]). However, in our research we focus on a subset of human activities that have the possibility of being temporally overlapped. We show that these types of activities can be recognized efficiently using our new framework.

3. Modeling Human Action

Most human activities are ordered hierarchically much like sentences in a natural language. Thus an understanding of hierarchy about human activities should be leveraged to reason about those activities , just like one might guess at the meaning of a word from its context. We assert that the SCFG and the BN lay the proper groundwork for hierarchical analysis of human activity recognition using a vision system.

Our justification in using a SCFG to model human activity is based on the idea that it models hierarchical structure that closely resembles the inherent hierarchy in human activity. Just as series of words can be represented at a higher level of abstraction, a series of primitive actions can also be represented at a higher level of abstraction. By recognizing the close analogy between a string of words and a series of actions, we reason that SCFGs are well suited for representing grammatical

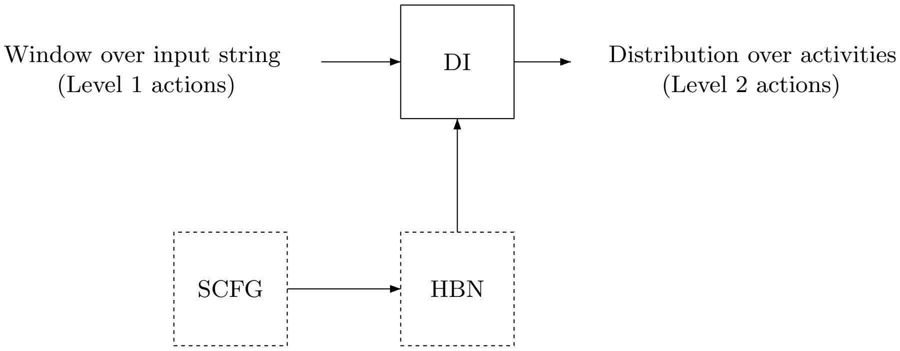


Fig. 1. System flow chart. Dashed lines indicate off-line components and solid lines indicate online components. Level 1 actions symbols and the HBN are merged via the deleted interpolation step to produce level 2 actions.

structure.

A SCFG is also able to describe an activity at any level in the hierarchy in the same way humans are known to perceive activities at different abstractions levels within a hierarchical structure. In contrast, standard sequential models like finite state machines, n-grams, Markov chains and hidden Markov models, do not explicitly model hierarchical structure.

Despite the expressive power of the SCFG, they were created to characterize formal language and thus in general, syntactic parsers are not well-suited for handling noisy data. Bayesian networks give us the robustness needed to deal with faulty sensor data, especially when dealing with human actions. In contrast to standard parsing algorithms, the merit of using an BN is found in the wide range of queries that can be executed over the network[18]. In addition, BNs can deal with negative evidence, partial observations (likelihood evidence) and even missing evidence, making it a favorable framework for vision applications.

4. Recognition System Overview

Our recognition system consists of three major parts (Figure 1). The first is the action grammar (a SCFG) that describes the hierarchical structure of all the activities to be recognized. Second is the hierarchical Bayesian network that is generated from the action grammar. Third is the final module that takes a stream of input symbols (level 1 action symbols) and uses deleted interpolation to determine the current probability distribution across each possible output symbol (level 2 action symbol).

We give the details of our system based on the use of the CAVIAR data set[19], which is a collection of video sequences of people in a lobby environment. The ground truth for each agent in each frame is labeled in XML with information about position, appearance, movement, situation, roles, and context. For practical reasons, we make direct use of the ground truth data to produce a sequence of primitive action symbols as the low-level input into our system.

Table 1. Grammar for producing level 1 symbols.

Level 1 Actions	Appearance	Movement	Position
en	appear	-	-
ex	disappear	-	-
ne	visible	active/walking	near exit/entrance
br	visible	active/inactive	near a landmark
in	visible	inactive	-
mp	visible	active	-
wa	visible	walking	-
pu	referenced to object properties		
pd	referenced to object properties		

4.1. *Action Grammar*

The set of terminals (level 1 action symbols) is defined as **T**= {*en, ne, ex, mp, wa, in, br, pu, pd* } (definitions given in Table 2). The level 1 action symbols were generated directly from the CAVIAR XML ground truth data using logical relationships between the appearance, movement and position information for each frame (Table 1). The set of action symbols (called level 2 actions) $\mathbf{A} = \{BI, BR, TK, LB, PT, AR, DP\}$, along with a set of intermediate action symbols $\mathbf{I} = \{AI, MV, MT, MF\}$ were created manually to be the set of high-level actions to be used by the system (Table 3). Level 2 actions are a special subset of nonterminal symbols in the level 2 grammar because they are direct abstraction productions of S (start symbol), i.e. they are directly caused by S. The set of nonterminals $\mathbf{N}$ is defined as $\mathbf{N} = \mathbf{I} \cup \mathbf{A}$. The set of production rules Σ and their corresponding probabilities are given in Table 4. We note here that since grammar creation is not the primary focus of our work, the grammar (including the rule probabilities) are manually defined.

4.2. *Hierarchical Bayesian Network*

We use a previously proposed method[18] to transform the action grammar (level 2 grammar) into a hierarchical Bayesian network (HBN). We use the term HBN because information about hierarchy from the SCFG is embedded in the BN.

As mentioned before, the SCFG is converted into a BN because it has the ability to deal with uncertainty. When the sensory input is uncertain, the BN can process

Table 2. Definition of the level 1 actions (terminal symbols).

Level 1 Actions	Meaning
en	enter : appears in the scene
ex	exit: disappears from the scene
ne	near exit/ entrance : moving near an exit / entrance
br	browse : standing near landmark
in	inactive: standing still
mp	move in place : standing but moving
wa	walk : moving within a certain velocity range
pd	put down : release object
pu	pick up : contact with object

Table 3. Definition of the level 2 actions and intermediate actions (nonterminal symbols).

Level 2 Actions	Meaning
AR	Arriving : Arriving into the scene
BI	Being Idle : Spending extra time in the scene
BR	Browsing : Showing interest in an object in the scene
TK	Taking away : Taking an object away
LB	Leaving behind : Leaving an object behind
PT	Passing Through : Passing through the scene
DP	Departing : Leaving the scene
Intermediate Actions	**Meaning**
AI	Action in Place: Taking action while in place
MV	Moving : Moving with a minimum velocity
MT	Move to : Moving in place after walking
MF	Move from : Walking after moving in place

a multinomial distribution across a discrete variable instead of a single value with a probability of one. In addition, the BN can also deal with missing evidence (a missed detection) by marginalizing over the values of the missed variable.

By converting the action grammar into a HBN, evidence nodes $\mathbf{E}$ contain terminal symbols, query nodes $\mathbf{Q}$ contain level 2 actions $\mathbf{A}$ and hidden nodes $\mathbf{H}$ contain production rules Σ or intermediate action $\mathbf{I}$. Results of transforming the grammar in Table 4 into a HBN is depicted in Figure 2.

We denote the probability density function (PDF) for level 2 actions[a] to be $\mathbf{P}(\mathbf{A}|\mathbf{e})$ where $\mathbf{A} = \{A_1, A_2, \ldots, A_v\}$ is the set of all level 2 actions (states). $\mathbf{e} = \{e_1, e_2, \ldots, e_l\}$ is a string of evidence at the evidence nodes of the HBN where l is the maximum length of the HBN. The probability of a specific level 2 action is

[a] $\mathbf{P}$ will be used when dealing with probabilities of multi-valued discrete variables. It denotes a set of equations with one equation for each value of the variable.

Table 4. Level 2 action grammar.

S → BI	0.20	BR → br	0.20
S → BR	0.10	BR → MV br	0.20
S → TK	0.05	BR → br mp	0.30
S → LB	0.05	BR → MV br mp	0.30
S → PT	0.30		
S → AR	0.15	LB → pd	0.50
S → DP	0.15	LB → MV pd	0.20
		LB → pd mp	0.05
BI → AI	0.10	LB → pd wa	0.05
BI → MV AI	0.10	LB → pd mp wa	0.10
BI → AI MV	0.10	LB → mp pd mp	0.10
BI → mp AI MV	0.10		
BI → mp	0.20	DP → ex	0.40
BI → MF mp	0.10	DP → wa ne ex	0.30
BI → MF	0.10	DP → ne ex	0.20
BI → MV ne MV	0.10	DP → wa ne	0.10
BI → AI wa ne	0.10		
		MV → MF	0.20
TK → pu	0.50	MV → MT	0.20
TK → MV pu	0.20	MV → wa	0.30
TK → pu mp	0.20	MV → mp	0.30
TK → pu wa	0.10		
TK → MV pu MV	0.10	MF → mp wa	1.00
		MT → wa mp	1.00
PT → en wa ex	0.70		
PT → ne wa ne	0.30	AI → in	0.60
		AI → br	0.20
AR → en	0.50	AI → pu	0.10
AR → en MV	0.50	AI → pd	0.10

defined as the sum of the probabilities from each of the query nodes,

$$\mathbf{P}(\mathbf{A}|\mathbf{e}) = \mathbf{P}(Q_1 = \mathbf{A}|\mathbf{e}) + \cdots + \mathbf{P}(Q_u = \mathbf{A}|\mathbf{e}). \tag{1}$$

When there are v different level 2 actions, $\mathbf{P}(\mathbf{A}|\mathbf{e})$ represents a set of v equations

$$\begin{aligned} P(A_1|\mathbf{e}) &= P(Q_1 = A_1|\mathbf{e}) + \cdots + P(Q_u = A_1|\mathbf{e}), \\ P(A_2|\mathbf{e}) &= P(Q_1 = A_2|\mathbf{e}) + \cdots + P(Q_u = A_2|\mathbf{e}), \\ \cdots \\ P(A_v|\mathbf{e}) &= P(Q_1 = A_v|\mathbf{e}) + \cdots + P(Q_u = A_v|\mathbf{e}). \end{aligned} \tag{2}$$

The probabilities of the level 2 actions $\mathbf{A} = \{A_1, A_2, \ldots, A_v\}$ always sum to one when the evidence can be explained by the grammar because $\mathbf{A}$ is the set of all

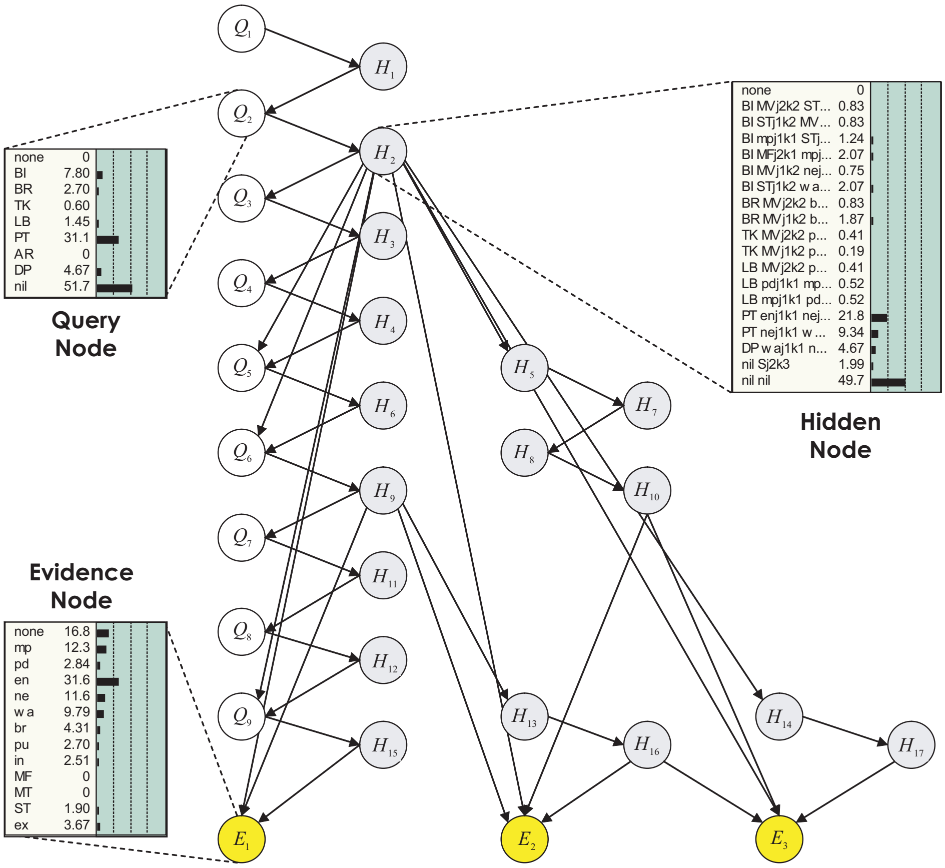


Fig. 2. Hierarchical Bayesian Network (maximum length $l = 3$). The content of each node type is depicted by a bar chart.

possible productions of S (start symbol). Thus,

$$\sum_{i=1}^{v} P(A_i|\mathbf{e}) = 1. \tag{3}$$

The computational cost of calculating the beta probabilities is $O(Pn^m d^m)$ and the cost of building the Bayesian network is $O(Pn^{m+1}d^mT^m)$ (more details in the original paper[18]). P is the number of rules induced by the grammar, d is the maximum number of abstraction levels, n is the maximum length of a sentential string, m is the maximum production length and T is the maximum number of entries of any conditional probability table in the network. Although the cost of building the network grows exponentially as the grammar grows in complexity the network only needs to be computed once offline. With respect to inference with the Bayesian network, exact inference becomes intractable as the network grows in size, which means that other well known approximation algorithms will need to be utilized for bigger networks.

4.3. *Deleted Interpolation*

The concept of deleted interpolation (DI) involves combining two (or more) models of which one provides a more precise explanation of the observations but is not always reliable and the another which is more reliable but not as precise. A precise model requires that the input data be a close fit to the model and will reject anything that does not match. A reliable model exhibits greater tolerance in fitting the data and is more likely to find a match. Combining models allows us to fall back on the more reliable model when the more precise model fails to explain the observations. It is called *deleted* interpolation because the models which are being interpolated use a subset of the conditioning information of the most discriminating function[20].

In our system we assume that the analysis of a long sequence of evidence is more precise than that of a shorter length because a long sequence takes into consideration more information. However, when analysis over a long (more precise) input sequence fails we would like to fall back on analysis based on a shorter (more reliable) subsequence.

To implement this we calculate the current probability distribution $\mathbf{S}$ across level 2 actions, at each time instance, as a weighted sum of models,

$$\mathbf{S} = \sum_{i=1}^{l} \lambda_i \mathbf{P}(\mathbf{A}|\mathbf{O}_i), \tag{4}$$

where $\mathbf{O}_i$ is the string of full evidence when $i = 1$ and represents smaller subsets of the evidence sequence as the index i increases. The weights are constrained by $\sum_{i=1}^{l} \lambda_i = 1$.

Representing our system as a dynamic Bayesian network yields the network in Figure 3. Memory nodes are added to the network to store past evidence and l is the length of the analysis window. When $l = 3$, the current probability distribution of the level 2 actions over the temporal window is given by the equation

$$\mathbf{S} = \lambda_1 \mathbf{P}(\mathbf{A}|\mathbf{O}_1) + \lambda_2 \mathbf{P}(\mathbf{A}|\mathbf{O}_2) + \lambda_3 \mathbf{P}(\mathbf{A}|\mathbf{O}_3), \tag{5}$$

where[b]

$$\mathbf{O}_1 = \{e_1^t, e_2^{t-1}, e_3^{t-2}\} \tag{6}$$

$$\mathbf{O}_2 = \{e_1^t, e_2^{t-2}, e_3^{none}\} \tag{7}$$

$$\mathbf{O}_3 = \{e_1^t, e_2^{none}, e_3^{none}\}. \tag{8}$$

The first term $\mathbf{P}(\mathbf{A}|e_1^t, e_2^{t-1}, e_3^{t-2})$ is the activity probability distribution of the complete set of evidence and represents activities that have started at $t - 2$. The

[b] e^{none} is a terminal symbol that represents an end of the sequence.

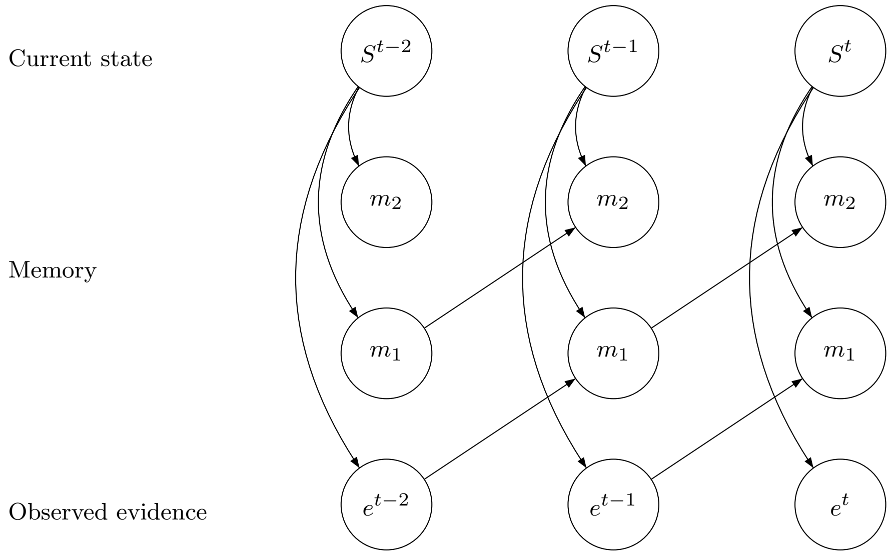


Fig. 3. System depicted as a dynamic Bayesian network where the memory elements store past evidence.

second term $\mathbf{P}(\mathbf{A}|e_1^t, e_2^{t-2}, e_3^{none})$ is the activity probability distribution for a partial set of evidence and represents activities starting at $t-1$. Likewise, the last term $\mathbf{P}(\mathbf{A}|e_1^t, e_2^{none}, e_3^{none})$ is a probability distribution for activities starting at t. This is the mechanism that effectively allows the system to represent overlapped activities.

5. Experimental Results

The following experiments show that our method is well-suited for recognizing sequential and overlapped single-agent activities. In the first two experiments we show the advantage of using DI as opposed to not using DI. In the latter two sections, the effect of the values chosen for grammar rule probabilities and the mixture weights are examined.

The video data used for this experiment was taken in a lobby environment (Figure 4) and the sequence of level 1 actions were generated using the labelled CAVIAR data. Analysis was run on six video sequences (Walk1, Walk2, Browse1, Browse2, Leave1 and Leave2) to test the performance of the system. The recognition results are depicted as stacked area graphs for each type of activity and are shown in Figures 5, 6 and 7. In each figure, the ground truth is given along with the results for each of the four different experimental setups.

Probabilistic inference with the BN was performed using an exact algorithm (junction tree) with Netica[21] for all of the experiments. However, as mentioned

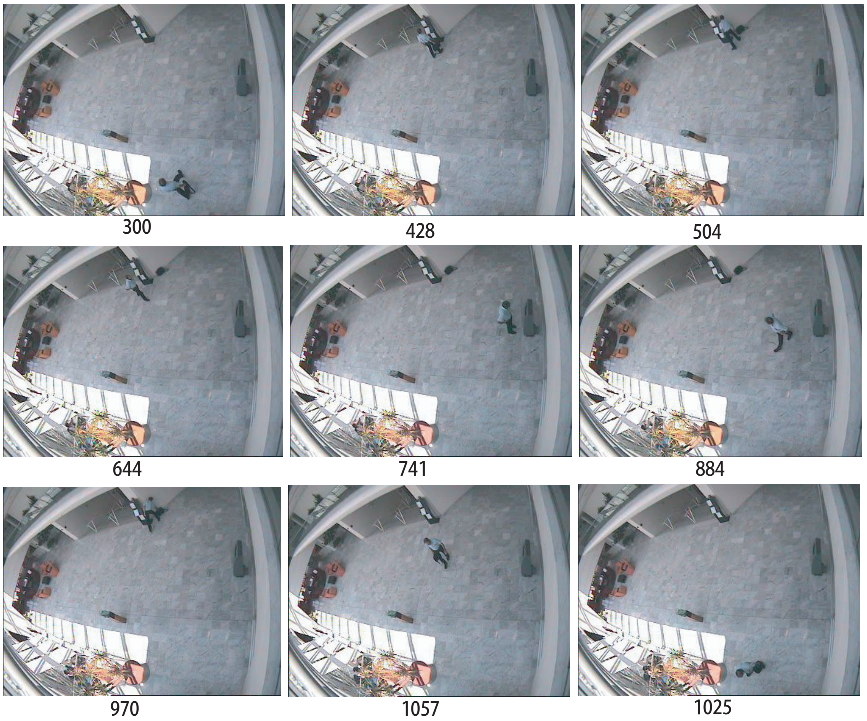


Fig. 4. Key frames for the ”Leave Behind and Pick Up” (Leave1) sequence.

previously, as the size of the grammar (and the BN) increases, approximation algorithms will need to be used to perform the inference task.

5.1. *Ground Truth*

The ground truth was compiled from multiple users, as a normalized sum of the interpretations of the video data. Each labeler was given a definition (Table 5) for each level 2 action and directed to label every frame for each action independently. Users were given the option of labeling each frame with a *yes*, *maybe* or *no* (10 points for *yes*, 5 points for *maybe* and 0 points for *no*). No restrictions were placed on the number of times they could re-label or review the video sequences. They were not shown the string of primitive symbols extracted from the CAVIAR data.

5.2. *Using Deleted Interpolation*

The recall rate, precision rate, miss rate and false detection rates are given for each of the six video sequences in Table 8 when deleted interpolation was implemented using the grammar in Table 4. The definition of each rate is given in Table 6.

The precision rate was 88% after filtering out a common problem (explained

Table 5. Definitions for ground truth labeling.

Arriving	A period of time where the agent has just entered the scene. It must occur near a known exit or entrance.
Passing Through	Agent appears to be simply walking through the lobby. Pattern should look like: Enter + passing through + exit. Agent is not looking around.
Being Idle	The agent appears to be walking around aimlessly. Usually characterized by walking slowly and stopping in place. Sometimes includes browsing.
Browsing	Period of time where the agent is near a landmark (counter, magazine rack, information booth). The agent appears to be looking at a landmark.
Taking Away	Agent appears to be picking something up or preparing to pick something up. Includes movement just before and after picking up the object.
Leaving Behind	The agent appears to be putting something down or preparing to put something down. Includes movement just before and after putting down the object.
Departing	Period of time where it seems that the agent is about to exit the scene. Ceases once the agent exits the scene.

Table 6. Definitions (a) Definition of the data types (b) Formulas for the different rates.

A	Number of RELEVENT documents RETRIEVED
B	Number of RELEVENT documents NOT RETRIEVED
C	Number of IRRELEVENT documents RETRIEVED
D	Number of IRRELEVENT documents NOT RETRIEVED

(a)

Recall: A/(A+B)	Relevant data retrieved from relevant data
Precision: A/(A+C)	Relevant data retrieved from retrieved data
Miss: B/(A+B)	Relevant data missed (1-recall)
False: C/(C+D)	Irrelevant data retrieved from all irrelevant data

(b)

later). *Arriving* and *Departing* had the highest precision rate (∼95%) because the activities were highly dependent on location (i.e. near a known exit or entrance) which made early detection relatively easy. In contrast, *Taking Away* had the lowest precision rate because the system was only able to detect the activity after the removed item was detected visually.

The frequent mis-detection of *Being Idle* as *Passing Through* had a negative effect on four of the six sequences, contributing to a 16% drop in the precision rate (Browse1, Browse2, Leave1 and Leave2). This drop in performance can be explained by the fact that the ground truth was collected under conditions that differ from our system. Under normal conditions, a system cannot know if an agent will become idle or not and therefore can only label an initial detection of a moblie agent, as *Passing Through* the scene. In contrast, the ground truth was labelled with the foreknowledge of what the agent would do in the subsequent frames, giving the user the ability to mark an agent as being idle upon entry into the scene. Therefore, we are justified in removing the mis-detection of *Being Idle* as *Passing Through* to obtain a more realist precision rate.

The recall (capture) rate was 59% (equivalently, a miss rate of 41%) which indicates that the system was not able to detect the activity for the complete duration of the level 2 action as described by the ground truth data. The low recall rate is caused by similar reasons stated for the precision rate. The foreknowledge of the entire sequence gave the labeller the ability to recognize activities much earlier than the visual information permits. In contrast, the system changes its output only when a new terminal symbol (a significant visual change) is encountered.

The false alarm rate was 3% (not including the effects of *Passing Through*). The low false alarm rate is expected because the input symbols (level 1 actions) only change when there is a significant change in an agents visual characteristics.

An example of the detection of overlapping (concurrent) activities can be seen in the first transition from *Passing Through* to *Departing* in Figure 5(c). At about frame 315, both activities are detected and depicted as two stacked regions. Similar detections of overlapped activities are observed for *Browsing* and *Being Idle* in Figure 6(c).

In comparison to the ground truth, is was observed that the transitions between activities were abrupt for the experimental results. The sharp transitions can be attributed to the fact that the input into the system was a discrete sequence of primitive actions (level 1 actions), where each symbols was output only when a significant visual change was detected in appearance and movement (as defined by the CAVIAR data). In contrast, the ground truth was based on more detailed visual queues (e.g. body posture, head position) and foreknowledge of the entire sequence. The ground truth was also averaged over the labels of multiple users, which allowed the transition between activities to become smoother.

5.3. *No Deleted Interpolation*

To understand the advantage of using DI, an experiment was performed again on the same sequences but without the use of DI. The removal of DI is equivalent to the use of a single HBN shifted over time over a fixed temporal window to recognize activities. Since subsequences of the evidence are not used to interpolate the results, several level 2 actions based on smaller strings were not detected by the system.

The level 2 action that was effected the most was *Departing* because the sequence of primitive symbols {*wa, ne, ex*} was never detected by the input sequences. Furthermore, since *Departing* relies heavily on the use of smaller substrings of one or two level 1 action symbols to detect, removing the DI framework significantly reduces the systems ability to recognize departures. In contrast, one instance of temporal concurrence was detected in Figure 7(b) between *Being Idle* and three other activities. This overlap was captured because in the grammar, a subset of the sequences of actions used by *Being Idle* was also used for the recognition of *Browsing*, *Leaving Behind* and *Taking Away*.

5.4. *DI using Uniformly Distributed Grammar Parameters*

The original grammar parameters (production rule probabilities) were set at the discretion of a knowledge engineer, giving greater weight to sequences that were more likely to occur. However, in this set of experiments, the probabilities were distributed uniformly among all possible productions for each nonterminal symbol. That is, the production probabilities $P(N \rightarrow \zeta_i)$ were uniformly distributed such that for the nonterminal N, $\sum_i P(N \rightarrow \zeta_i) = 1$ where ζ is a string of one or more symbols on the right-hand side of the production rule.

Since changing the probabilities of the rules changes only the proportions between overlapped activities and not the duration of activities themselves, the rates remain the same. It is interesting to observe that the proportion of the probabilities between activities remain virtually unchanged after rule probabilities have been changed (Figures 5(d), 6(d) and 7(d)). This is due to the fact that the structural analysis of a symbol sequence plays a larger role in determining the results compared to the role of the probabilities of the rules. Therefore, it is more important to include the correct rules in the grammar than to assign the optimal probabilities.

5.5. *DI using Uniformly Distributed Mixture Weights*

Previously, the mixture weights for deleted interpolation were set so that $\lambda_1 > \lambda_2 > \cdots > \lambda_l$, giving more weight to longer subsets of the data. For this experiment, the mixture weights λ_i were uniformly distributed, giving equal weight to each term in the interpolation equation. That is, $\sum_{i=1}^{l} \lambda_i = 1$ and $\lambda_i = 1/l$. A uniform weighting scheme can be interpreted as giving equal confidence to each of the l terms in the DI equation.

Small changes in the proportions between overlapped probabilities were observed for the detection of *Departing* and *Passing through* (a higher probability for *Departing*), which was closer to the ground truth. In general however, the results remained similar to the results of using the original weighting scheme (Figures 5(e), 6(e) and 7(e)). As in the previous experiment, we observe here that the structural constraints outweigh the values of the mixture weights such that the proportions between overlapped activities change only nominally when the mixture weights are varied. Again, since the mixture weights only effect the proportion between the probabilities of the actions and not their durations, the detection rates remain unchanged.

6. Summary and Conclusion

We have addressed the issue of hierarchical representation of human activity by basing our system on a SCFG. We then converted the SCFG to a HBN to allow the system to make complex probabilistic queries needed for uncertain inputs. As a preliminary test, we then applied the HBN using DI to discover overlapped activities over a string of discrete primitive action symbols. Through a set of preliminary experiments, it was shown that our methodology is well-suited for detecting the overlap of simple single-agent activities.

We recognize that manually defining the grammar may be problematic when a pre-defined grammar is not available or when the input string is noisy. In another work, we develop an unsupervised grammar learning technique to extract a grammar from a noisy input sequence generated from a real video sequence[22].

The input sequence for this work was a discrete string of action symbols. However, in reality human actions are continuous and vision-based methods for detecting human actions are usually uncertain and noisy. Future work will focus on dealing with more realistic inputs under a similar hierarchical framework.

References

1. Jeffrey M. Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127:3–21, 2001.
2. Toshikazu Wada and Takashi Matsuyama. Appearance based behavior recognition by event driven selective attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–764, 1998.
3. Douglas Ayers and Mubarak Shah. Monitoring human behavior from video taken in an office environment. *Image Vision Comput.*, 19(12):833–846, 2001.
4. Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385. IEEE Computer Society, 1992.
5. Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.
6. Aphrodite Galata, Neil Johnson, and David Hogg. Learning variable-length Markov models of behavior. *Computer Vision and Image Understanding*, 81(3):398–413, 2001.
7. Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. In *Proceedings of the IEEE International Conference on Multimodal Interfaces*, pages 3–8. IEEE Computer Society, 2002.
8. Shaogang Gong and Tao Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 742–749. IEEE Computer Society, 2003.
9. Xiaohui Liu and Chin-Seng Chua. Multi-agent activity recognition using observation decomposed hidden Markov model. In *Proceedings of the Third International Conference on Computer Vision Systems*, pages 247–256, 2003.
10. Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,

pages 838–845, 2005.

11. Matthew Brand. Understanding manipulation in video. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, page 94, 1996.
12. Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.
13. Darnell J. Moore and Irfan A. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proceedings of the National Conference on Artificial Intelligence*, pages 770–776, 2002.
14. David Minnen, Irfan A. Essa, and Thad Starner. Expectation grammars: Leveraging high-level expectations for activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II: 626–632, 2003.
15. Hung Hai Bui, Svetha Venkatesh, and Geoff A. W. West. Tracking and surveillance in wide-area spatial environments using the abstract hidden Markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):177–195, 2001.
16. Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.
17. Nam T. Nguyen, Hung Hai Bui, Svetha Venkatesh, and Geoff A. W. West. Recognising and monitoring high-level behaviours in complex spatial environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II: 620–625, 2003.
18. David V. Pynadath and Michael P. Wellman. Generalized queries on probabilistic context-free grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):65–77, 1998.
19. EC funded CAVIAR project under the IST fifth framework programme (ist-2001-37540). Found at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.
20. Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2003.
21. Netica. Found at http://www.norsys.com/.
22. Kris M. Kitani, Yoichi Sato, and Akihiro Sugimoto. Recovering the basic structure of human activities from a video-based symbol string. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 9–9, 2007.

Table 7. Counts for the different rates.

	COUNT	WALK1	WALK2	BROWSE1	BROWSE2	LEAVE1	LEAVE2	Total
Arriving	A	52	144	87	90	116	38	527
	B	3	129	42	50	49	3	276
	C	8	0	0	7	2	13	30
	D	218	904	553	443	719	836	3673
Passing Through	A	215	569	0	0	0	0	784
	B	32	63	0	0	0	0	95
	C	0	48	202	320	263	143	976
	D	34	497	480	270	623	747	2651
Being Idle	A	0	0	360	63	229	645	1297
	B	0	0	211	346	208	158	923
	C	0	0	0	29	151	43	223
	D	281	1177	111	152	298	44	2063
Browsing	A	0	0	189	21	0	209	419
	B	0	0	65	112	0	155	332
	C	0	0	5	0	0	42	47
	D	281	1177	423	457	886	484	3708
Taking Away	A	0	0	0	0	82	12	94
	B	0	0	0	0	32	56	88
	C	0	0	0	0	76	0	76
	D	281	1177	682	590	696	822	4248
Leaving Behind	A	0	0	0	0	62	16	78
	B	0	0	0	0	27	47	74
	C	0	0	0	0	8	27	35
	D	281	1177	682	590	789	800	4319
Departing	A	26	158	20	48	151	45	448
	B	94	522	31	9	76	27	759
	C	0	0	16	0	1	0	17
	D	161	497	615	533	658	818	3282

Table 8. Rates for recall, precision, miss and false alarm.

	RATE	WALK1	WALK2	BROWSE1	BROWSE2	LEAVE1	LEAVE2	Average
Arriving	Recall	94.5%	52.7%	67.4%	64.3%	70.3%	92.7%	65.6%
	Precision	86.7%	100.0%	100.0%	92.8%	98.3%	74.5%	94.6%
	Miss	5.5%	47.3%	32.6%	35.7%	29.7%	7.3%	34.4%
	False	3.5%	0.0%	0.0%	1.6%	0.3%	1.5%	0.8%
Passing Through	Recall	87.0%	90.0%					89.2%
	Precision	100.0%	92.2%					44.5%
	Miss	13.0%	10.0%					10.8%
	False	0.0%	8.8%	29.6%	54.2%	29.7%	16.1%	26.9%
Being Idle	Recall			63.0%	15.4%	52.4%	80.3%	58.4%
	Precision			100.0%	68.5%	60.3%	93.8%	85.3%
	Miss			37.0%	84.6%	47.6%	19.7%	41.6%
	False	0.0%	0.0%	0.0%	16.0%	33.6%	49.4%	9.8%
Browsing	Recall			74.4%	15.8%	0.0%	57.4%	55.8%
	Precision			97.4%	100.0%	0.0%	83.3%	89.9%
	Miss			25.6%	84.2%	0.0%	42.6%	44.2%
	False	0.0%	0.0%	1.2%	0.0%	0.0%	8.0%	1.3%
Taking Away	Recall					71.9%	17.6%	51.6%
	Precision					51.9%	100.0%	55.3%
	Miss					28.1%	82.4%	48.4%
	False	0.0%	0.0%	0.0%	0.0%	9.8%	0.0%	1.8%
Leaving Behind	Recall					69.7%	25.4%	51.3%
	Precision					88.6%	37.2%	69.0%
	Miss					30.3%	74.6%	48.7%
	False	0.0%	0.0%	0.0%	0.0%	1.0%	3.3%	0.8%
Departing	Recall	21.7%	23.2%	39.2%	84.2%	66.5%	62.5%	37.1%
	Precision	100.0%	100.0%	55.6%	100.0%	99.3%	100.0%	96.3%
	Miss	78.3%	76.8%	60.8%	15.8%	33.5%	37.5%	62.9%
	False	0.0%	0.0%	2.5%	0.0%	0.2%	0.0%	0.5%

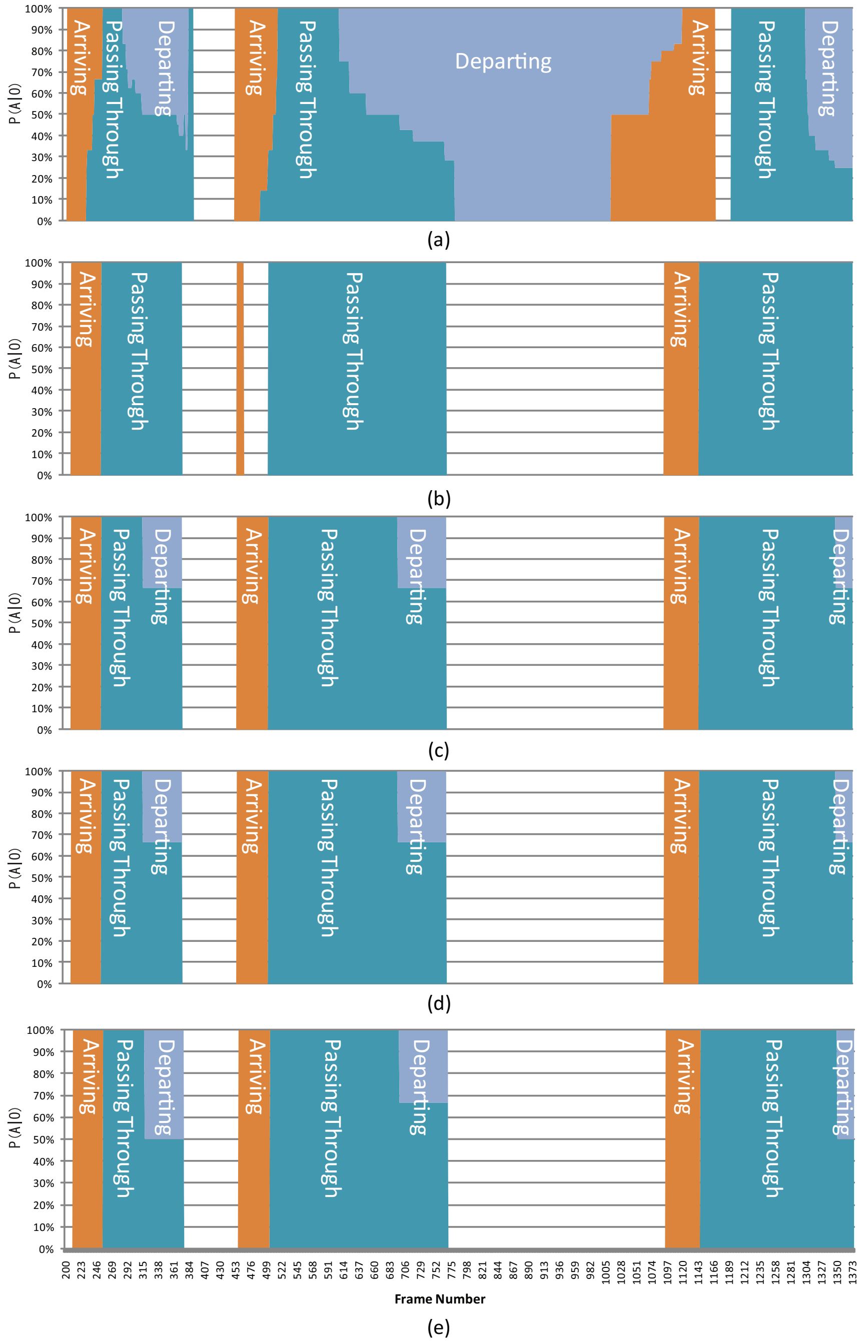
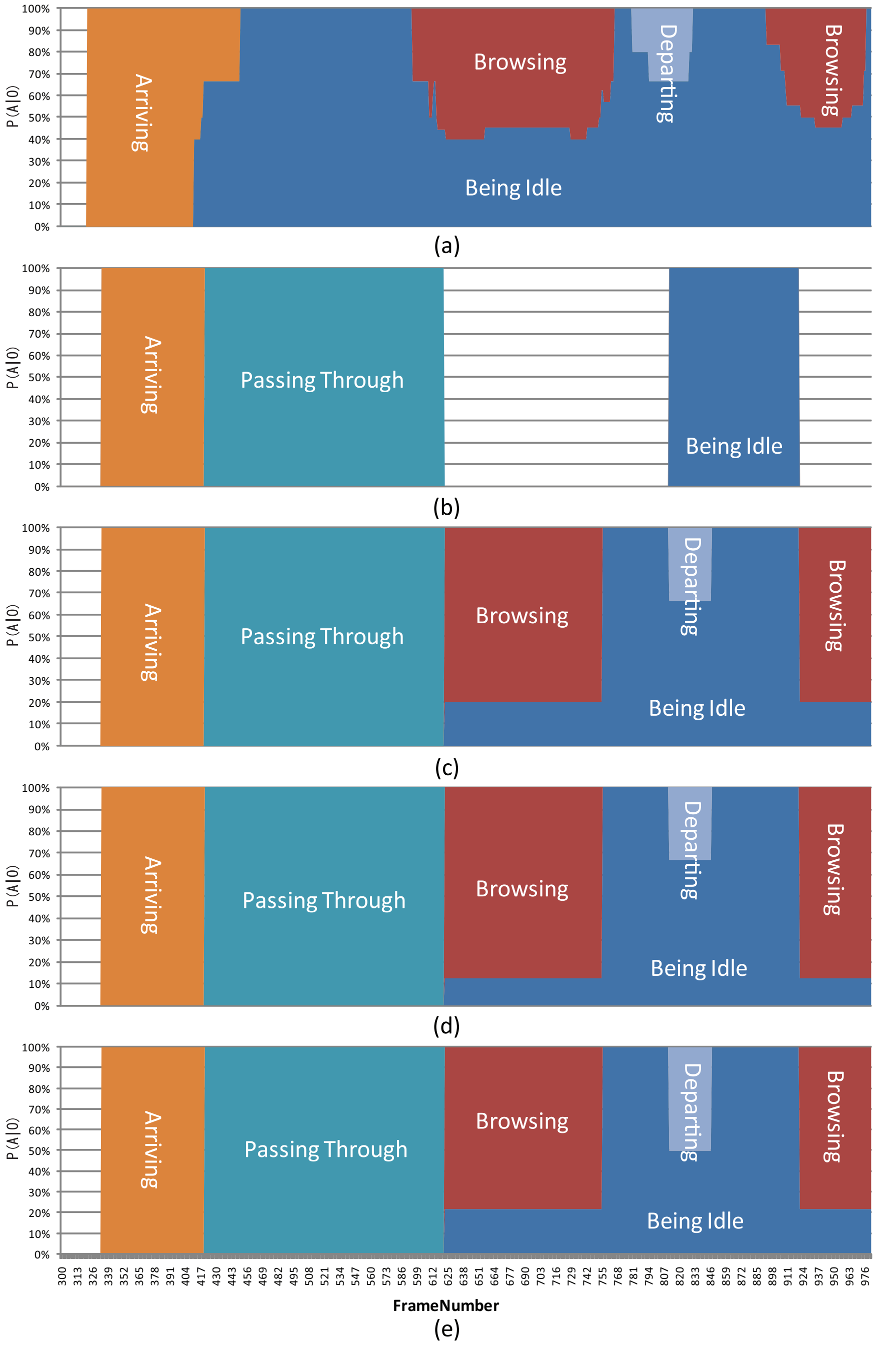


Fig. 5. Walk sequence (a) ground truth (b) no DI (c) DI with user defined rule probabilities (d) DI with uniformly distributed rule probabilities (e) DI with uniform mixture weights



(a)

(b)

(c)

(d)

(e)

Fig. 6. Browse sequence (a) ground truth (b) no DI (c) DI with user defined rule probabilities (d) DI with uniformly distributed rule probabilities (e) DI with uniform mixture weights

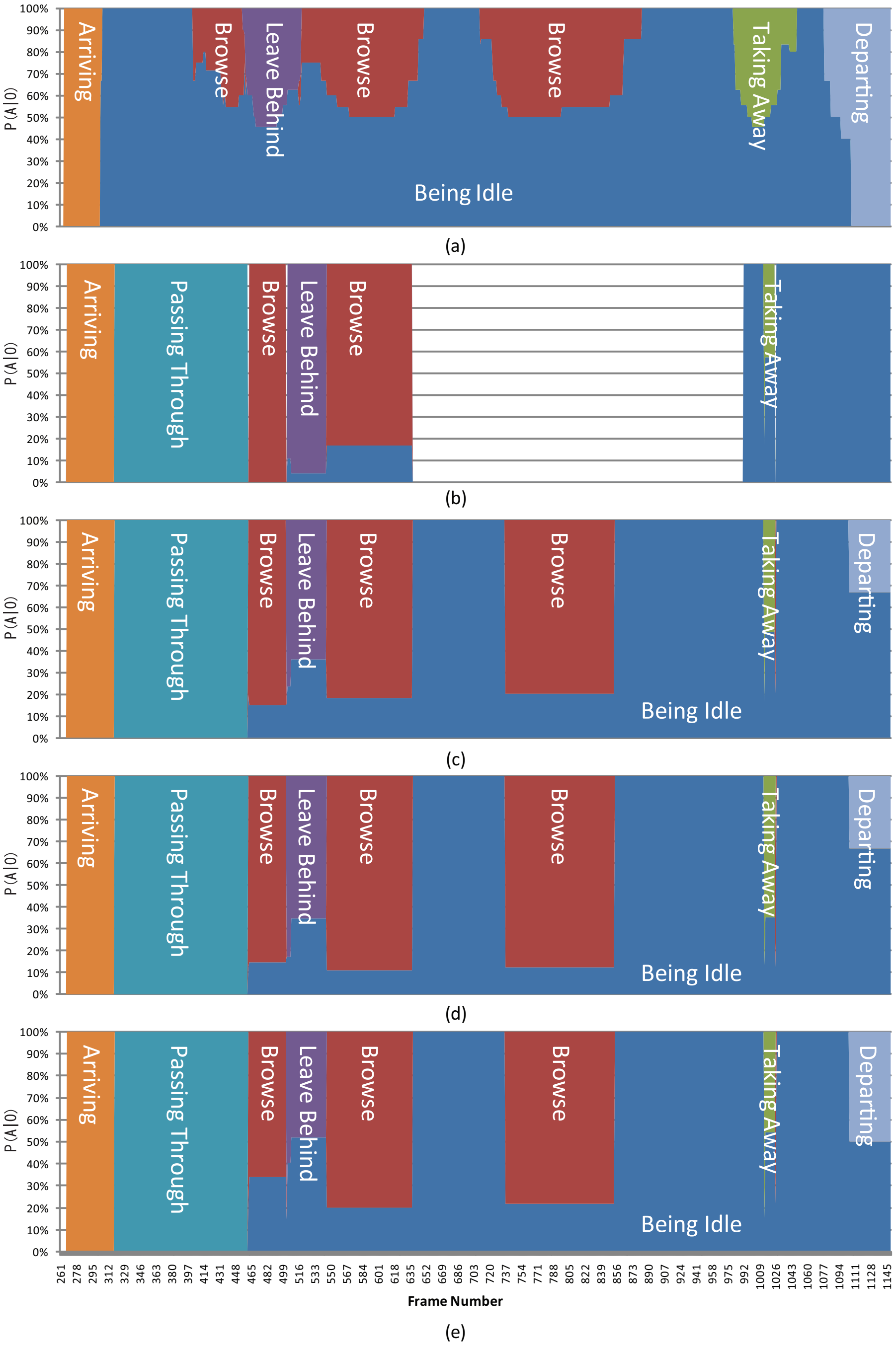


Fig. 7. Leave bag sequence (a) ground truth (b) no DI (c) DI with user defined rule probabilities (d) DI with uniformly distributed rule probabilities (e) DI with uniform mixture weights