Deleted Interpolation Using a Hierarchical Bayesian Grammar Network for Recognizing Human Activity

Kris M. Kitani, Yoichi Sato
Institute of Industrial Science
The University of Tokyo
Tokyo 158-8505, Japan
{kitani, ysato}@iis.u-tokyo.ac.jp

Akihiro Sugimoto
National Institute of Informatics
Tokyo 101-8430, Japan
sugimoto@nii.ac.jp

Abstract

From the viewpoint of an intelligent video surveillance system, the high-level recognition of human activity requires a priori hierarchical domain knowledge as well as a means of reasoning based on that knowledge. We approach the problem of human activity recognition based on the understanding that activities are hierarchical, temporally constrained and temporally overlapped. While stochastic grammars and graphical models have been widely used for the recognition of human activity, methods combining hierarchy and complex queries have been limited. We propose a new method of merging and implementing the advantages of both approaches to recognize activities in real-time. To address the hierarchical nature of human activity recognition, we implement a hierarchical Bayesian network (HBN) based on a stochastic context-free grammar (SCFG) . The HBN is applied to digressive substrings of the current string of evidence via deleted interpolation (DI) to calculate the probability distribution of overlapped activities in the current string. Preliminary results from the analysis of activity sequences from a video surveillance camera show the validity of our approach.

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 then becomes two-fold. First, we need a psychological framework for *characterizing* human activity and secondly, we need a computational method of *analyzing* those activities.

The characteristics of human activities can be learned from perceptual psychology [1]. Activities are hierarchical. That is, they are taxonomically 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 constrained. 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 been created heuristically.

2 Related Research

Contributions from computer vision and pattern recognition started with Brand in [2] and [3], 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 [4] 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 [5] 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 [6] 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 [7] used a hierarchy of abstract policies using Abstract Hidden Markov Models (AHMM) implementing a probabilistic state-dependent grammar to recognize action. The system recognizes people going to the library and using the printer across multiple rooms. AHMMs closely resemble the Hierarchical Hidden Markov Models (HHMM) [8] but with an addition of an extra state node. Nguyen [9] used an abstract Hidden Memory Markov Model (AHMEM), a modified version of the AHMM, for the same scenario as Bui.

The aforementioned works use 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 [4]). However, in our research we focus on a subset of human actions that have the possibility of being temporally overlapped. We show that these types of actions can be recognized efficiently using our new framework.

3 Modeling Human Action

Most human actions are ordered hierarchically much like sentences in a natural language. Thus an understanding of hierarchy about human actions should be leveraged to reason about those actions, just like one might guess at the meaning of a word from its context. We assert that the SCFG [10] and the BN [11] 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 strings of words and series of actions, we reason that SCFGs are well suited for representing grammatical structure.

Despite the expressive power of the SCFG, they were created to characterize formal language and thus in general, parsers are not well-suited for handling uncertainty. 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, as explained in [12]. 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 for 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 [13], 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 as low-level input into our system and in the next subsection we explain how this ground truth data was used to create an input stream for our system.

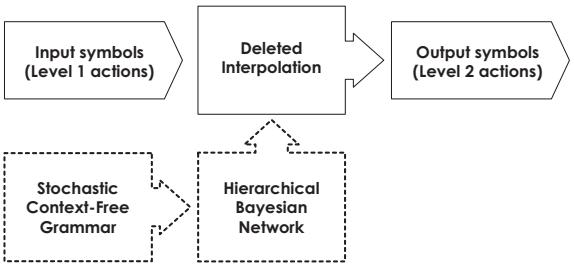

Figure 1: System Flow Chart. Dashed lines indicate off-line components and bold lines indicate online components. Level 1 actions symbols and the HBN are merged via the deleted interpolation step to produce level 2 actions.

Table 1: Definition of the grammar symbols (a) Grammar for producing level 1 symbols (b) Definition of the level 1 actions (terminal symbols) (c) Definition of the level 2 actions and intermediate actions (nonterminal 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		

(a)

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

(b)

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

(c)

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* } (see Figure 1b). The level 1 action symbols were generated directly from the CAVIAR XML ground truth data using the *appearance*, *movement* and *position* information for each frame (Figure 1a) . [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 a set of high-level actions to be used by the system (see Figure 1c). 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

[1]Again, we note here that neither grammar creation nor the extraction of low-level symbols is the focus of our work. Instead, we assume a set of reliable input symbols and present a hand-given grammar to validate our system.

Table 2: Level 2 action grammar.

Production Rule	Probability	Production Rule	Probability
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

N is defined as $\mathbf{N} = \mathbf{I} \cup \mathbf{A}$. The set of production rules Σ and their corresponding probabilities **p** are given in Table 2.

4.2 Hierarchical Bayesian Network

We use the methods presented in [12] 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. By converting the action grammar to a HBN, evidence nodes **E** contain terminal symbols, query nodes **Q** contain level 2 actions **A** and hidden nodes **H** contain production rules Σ or intermediate action **I**. Results of transforming the grammar in Table 2 into a HBN is depicted in Figure 2.

We denote the probability density function (PDF) for level 2 actions[2] to be $\mathbf{P}(\mathbf{A}|\mathbf{e})$ where $\mathbf{A} = \{A_1, A_2, \ldots, A_n\}$ 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 defined as the sum of the probabilities from each of the query nodes,

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

[2]**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.

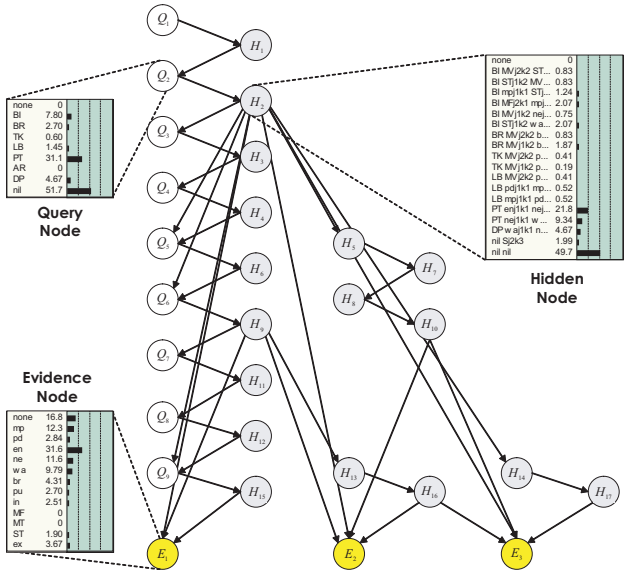

Figure 2: Hierarchical Bayesian Network (maximum length $l = 3$). The content of each node type is depicted by a bar chart.

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

$$\begin{aligned} P(A_1|\mathbf{e}) &= P(Q_1 = A_1|\mathbf{e}) + \cdots + P(Q_m = A_1|\mathbf{e}), \\ P(A_2|\mathbf{e}) &= P(Q_1 = A_2|\mathbf{e}) + \cdots + P(Q_m = A_2|\mathbf{e}), \\ &\cdots \\ P(A_n|\mathbf{e}) &= P(Q_1 = A_n|\mathbf{e}) + \cdots + P(Q_m = A_n|\mathbf{e}). \end{aligned}$$

The probabilities of the level 2 actions $\mathbf{A} = \{A_1, A_2, \ldots, A_n\}$ always sum to one when the evidence can be explained by the grammar because $\mathbf{A}$ is the set of all possible productions of S (start symbol). Thus,

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

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 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 [10].

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.

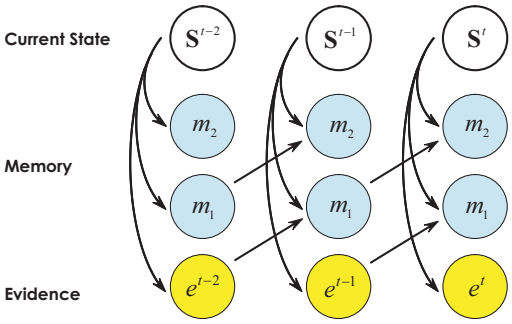

Figure 3: System depicted as a dynamic Bayesian network where the memory elements store past evidence.

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,

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

where $\mathbf{O}_i$ is the string of full evidence when $i = 1$ and represents digressive subsequences of the evidence as the index i increases. The weights are constrained by $\sum_{i=1}^{l} \lambda_i = 1$. At present the weights are hand-given but for future works, weights will be estimated using expected maximization or some other numerical algorithm.

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 the maximum number of memory nodes is $l - 1$, where l is the length of the analysis window. When $l = 3$ the current probability distribution of the level 2 actions over a temporal window of 3 is given by the equation

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

where [3]

$$\begin{aligned} \mathbf{O}_1 &= \{e_1^t, e_2^{t-1}, e_3^{t-2}\} \\ \mathbf{O}_2 &= \{e_1^t, e_2^{t-2}, e_3^{none}\} \\ \mathbf{O}_3 &= \{e_1^t, e_2^{none}, e_3^{none}\}. \end{aligned}$$

In the special case of using DI for activity recognition, each term in the DI equation represents the probability distribution for a new activity (level 2 action) beginning at a previous time instance. The resulting current probability distribution $\mathbf{S}$ is a mixture of probability distributions of various overlapping activities. This is the mechanism that effectively allows the system to represent overlapped activities.

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

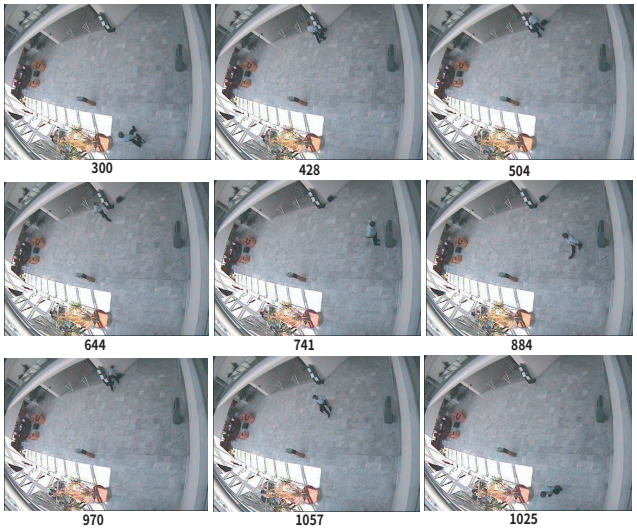

Figure 4: Key frames for the "Leave Behind and Pick Up" (Leave1) sequence.

5 Experimental Results

The following experiment shows that our method for single-agent activity recognition is well-suited for recognizing temporally constrained and temporally overlapped activities.

The ground truth was compiled as a normalized sum of the interpretations of multiple people. Each labeler was given a definition (shown in Table 3) for each level 2 action and asked to label each action separately. They were also 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 the video sequences.

Analysis was run on six video sequences (Walk1, Walk2, Browse1, Browse2, Leave1 and Leave2) to test the detection rate of the system. The results for each of the six video sequences are shown as normalized stacked area charts. Temporal overlap between level 2 actions are depicted as a decreasing or increasing stair case. All of the seven level 2 actions contained in the six sequences were successfully detected using the proposed system.

Figure 5 shows an example of a temporally overlapped activity in the Walk1 sequence. The upper chart depicts the results of the ground truth labeling showing a transition from *Arriving* to *Passing Through* to *Departing*. Although the system was not able to reproduce the transition from *Arriving* (light blue) to *Passing Through* (purple), the overlap of *Passing Through* (purple) to *Departing* (dark blue) is depicted from about frame 480 to the end. Similar results were acquired for the remaining 5 sequences (Figure 6, Figure 7, Figure 8, Figure 9 and Figure 10). A summary of the

Table 3: 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 stoping 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 4: Definitions (a) Definition of the data types (b) Formulas for the different rates.

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

(a)

RECALL : A / (A + B)	Relevant data retrieved from all relevant data
PRECISION : A / (A + C)	Relevent data retrieved from all retrieved data
MISS : B / (A + B)	Relevant data missed (1 - Recall)
FALSE ALARM : C / (C + D)	Irrelevant data retrieved from all irrelevant data

(b)

quantitative results for each sequence is given in Table 5.

The precision rate was 88% after filtering out a common problem (explained later). Including all the data, the overall precision rate was 72% which means that the detection of level 2 actions happened within the duration of the true action 72% of the time. *Arriving* and *Departing* had the highest precision rate ($\sim$95%) because the activities were highly dependent location (i.e. near a known exit or entrance). Conversely, *Taking Away* had the lowest precision rate because while the ground truth data indicated an early starting point for the activity, the system was only able to detect the activity once the item was actually detected visually. The frequent misinterpretation of *Being Idle* as *Passing Through* had a negative effect on four out of the six sequences, contributing to a 16% drop in the precision rate (Browse1, Browse2, Leave1 and Leave2). Filtering out the misinterpretations of *Passing Through* (which is justified since a labeler without foreknowledge would have given similar results) would increase the overall precision rate to 88%.

The recall (capture) rate was 59% (equivalently, a miss

Table 5: Summary (a) Counts for the different rates (b) Rates for recall, precision, miss and false alarm.

	Data Type	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**

(a)

	Rate	WALK1	WALK2	BROWSE1	BROWSE2	LEAVE1	LEAVE2	Averaae
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 Alarm	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 Alarm	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 Alarm	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 Alarm	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 Alarm	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 Alarm	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 Alarm	0.0%	0.0%	2.5%	0.0%	0.2%	0.0%	**0.5%**

(b)

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. This can be explained by the nature of the input data which only changes states when there is a significant change in the movement and appearance of the agent (i.e. when a new terminal symbol is encountered) whereas the labeler uses a wider set of knowledge to reason about the actual duration of the activity. The gradual transitions between level 2 actions could be better characterized by utilizing likelihood probabilities (probabilities associated with each terminal symbol) for each piece of evidence.

The false alarm rate was 6% but drops to 3% when the effects of *Passing Through* are removed. The low false alarm rate is expected because the input symbols (level 1 actions) only change when there is a true change in an agents physical state. However, in light of future modifications to improve the recall rate and precision rate, the introduction of likelihood probabilities might have an adverse effect on the false alarm rate.

A simulation of the effects of a faulty vision sensor can be seen in the Walk2 sequence (Figure 6). While the ground truth data shows that *Departing* is continuously detected between frames 800 and 1000, the analysis results show a big gap in the sequence. This can be explained by the fact that from about frame 784 to frame 1016, the agent is moving around in a dark area near the back of the room. Although the agent is still perceived by the human eye, the location of the agent is unknown to the low-level vision system (i.e. simulates a lack of detection by a vision system). Thus, the system assumes that the agent has completely departed in frame 771. However, at frame 1096 the system encounters a *enter* symbol and recognizes the agent to be *Arriving* to the scene.

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 vision applications. We then applied the HBN using DI to discover overlapped activities in the input string of 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.

In regards to the entire system, there are two challenges that will need to be addressed in future research, namely: the method of grammar creation and a framework for activity prediction. Manual grammar creation is not practical for large grammars and inversely, it is very difficult for a system to learn high-level activities without any supervision. Future work will need to create an interface and computational framework to find the right balance between user defined grammars and grammars discovered by the system.

While the current framework also allows for activity prediction by marginalizing probabilities for future time steps, the conditions for activity prediction must still be evaluated. If a robust framework for activity prediction is found, predictive information could be used in many ways. Prediction about future activities using high-level reasoning would lead to a whole new realm of applications.

Acknowledgments

A part of this work was supported by Grants-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan (No. 13224051).

References

[1] Jeffrey M. Zacks and Barbara Tversky. Event Structure in Perception and Conception. *Psychological Bulletin*, 127:3–21, 2001.

[2] Matthew Brand. Understanding Manipulation in Video. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, page 94. IEEE Computer Society, 1996.

[3] Matthew Brand. The ”Inverse Hollywood Problem”: From Video to Scripts and Storyboards via Causal Analysis. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 132–137, 1997.

[4] 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.

[5] Darnell J. Moore and Irfan A. Essa. Recognizing Multitasked Activities from Video Using Stochastic Context-Free Grammar. In *Eighteenth national conference on Artificial intelligence*, pages 770–776, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[6] David Minnen, Irfan A. Essa, and Thad Starner. Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II: 626–632. IEEE Computer Society, 2003.

[7] 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.

[8] Shai Fine, Yoram Singer, and Naftali Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1):41–62, 1998.

[9] 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 Computer Society Conference on Computer Vision and Pattern Recognition*, pages II: 620–625. IEEE Computer Society, 2003.

[10] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2003.

[11] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[12] 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.

[13] EC funded CAVIAR project under the IST Fifth Framework Programme (IST-2001-37540). Found at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

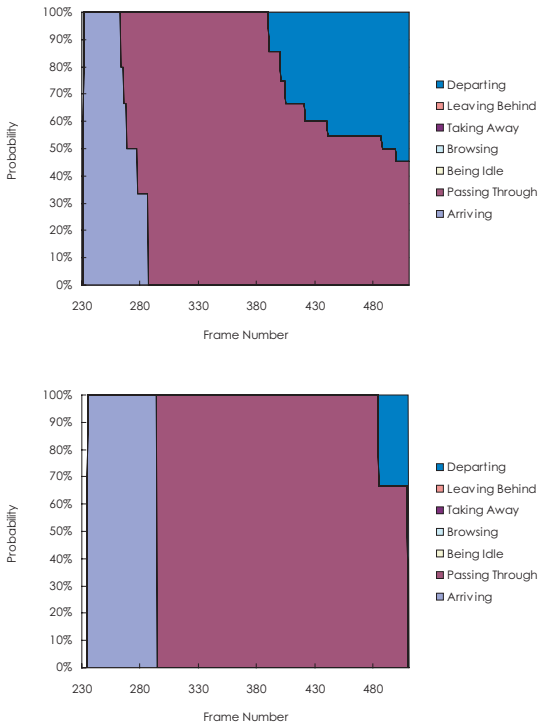

Figure 5: Walk1 : Ground truth (top) and analysis results (bottom).

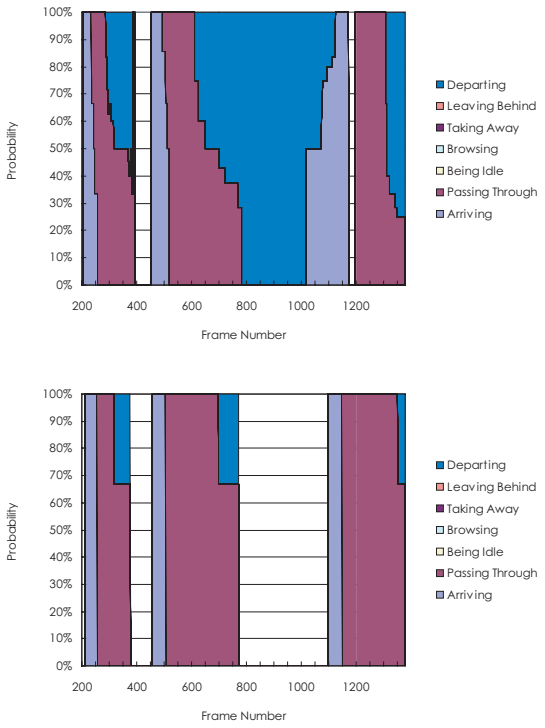

Figure 6: Walk2 : Ground truth (top) and analysis results (bottom). Simulates the effect of a faulty vision system.

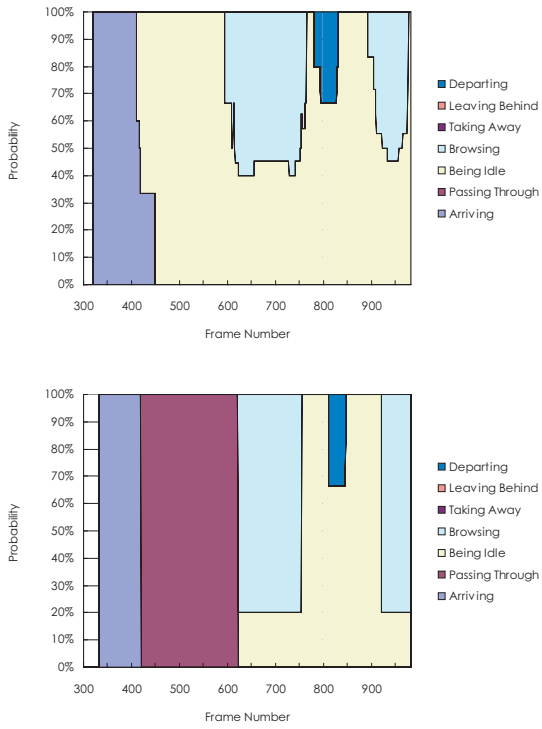

Figure 7: Browse1 : Ground truth (top) and analysis results (bottom).

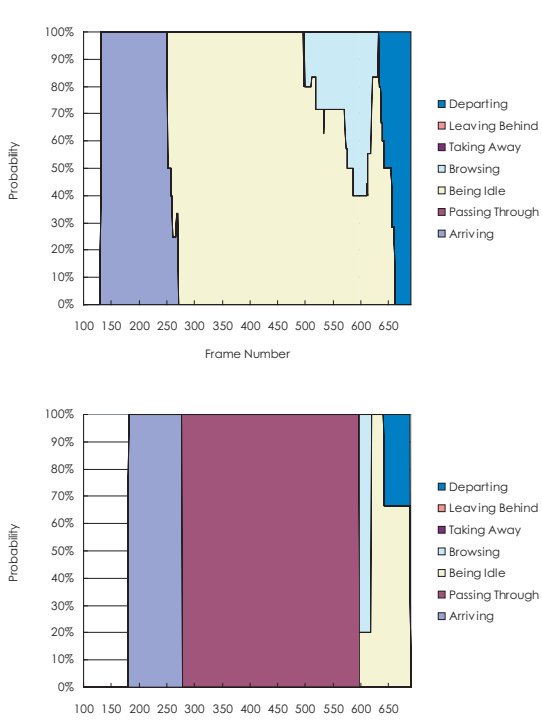

Figure 8: Browse2 : Ground truth (top) and analysis results (bottom).

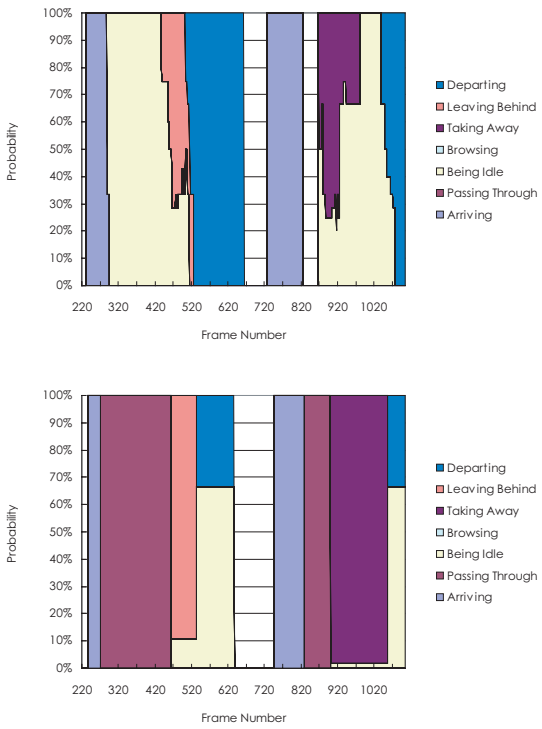

Figure 9: Leave1 : Ground truth (top) and analysis results (bottom).

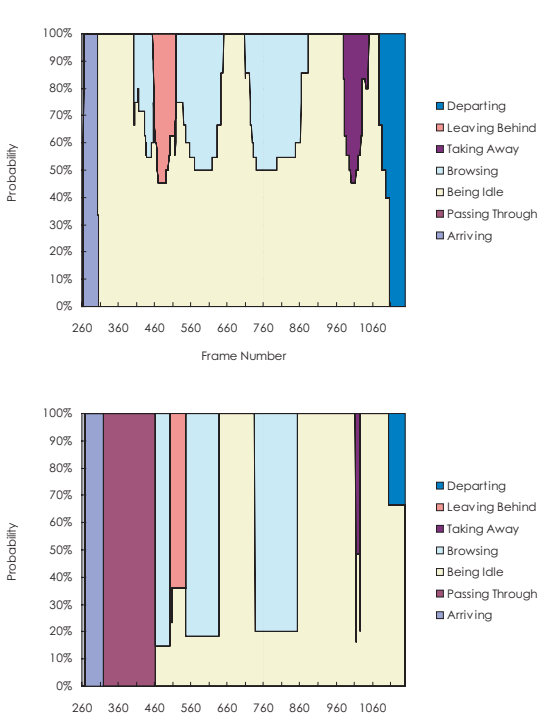

Figure 10: Leave2 : Ground truth (top) and analysis results (bottom).