

 Open access • Proceedings Article • DOI:10.1109/CVPR.2003.1211524

## Recognizing and monitoring high-level behaviors in complex spatial environments

— [Source link](#) 

N.T. Nguyen, Hung Bui, S. Venkatsh, Geoff West

**Institutions:** Curtin University

**Published on:** 18 Jun 2003 - Computer Vision and Pattern Recognition

**Topics:** Hidden Markov model

Related papers:

- [Recognition of visual activities and interactions by stochastic parsing](#)
- [A tutorial on hidden Markov models and selected applications in speech recognition](#)
- [Activity recognition and abnormality detection with the switching hidden semi-Markov model](#)
- [Layered representations for human activity recognition](#)
- [Learning patterns of activity using real-time tracking](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/recognizing-and-monitoring-high-level-behaviors-in-complex-50paton5tx>

# Deakin Research Online

## **This is the published version:**

Nguyen, Nam T., Bui, Hung H., Venkatesh, Svetha and West, Geoff 2003, Recognising and monitoring high-level behaviours in complex spatial environments, *in CVPR 2003 : Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Los Alamitos, Calif., pp. 620-625.

## **Available from Deakin Research Online:**

<http://hdl.handle.net/10536/DRO/DU:30044640>

Reproduced with the kind permissions of the copyright owner.

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

**Copyright** : 2003, IEEE

# Recognising and Monitoring High-Level Behaviours in Complex Spatial Environments

Nam T. Nguyen   Hung H. Bui   Svetha Venkatesh   Geoff West  
School of Computing, Curtin University of Technology  
{nguyentn, buihh, svetha, geoff}@cs.curtin.edu.au

## Abstract

*The recognition of activities from sensory data is important in advanced surveillance systems to enable prediction of high-level goals and intentions of the target under surveillance. The problem is complicated by sensory noise and complex activity spanning large spatial and temporal extents. This paper presents a system for recognising high-level human activities from multi-camera video data in complex spatial environments. The Abstract Hidden Markov mEmory Model (AHMEM) is used to deal with noise and scalability. The AHMEM is an extension of the Abstract Hidden Markov Model (AHMM) that allows us to represent a richer class of both state-dependent and context-free behaviours. The model also supports integration with low-level sensory models and efficient probabilistic inference. We present experimental results showing the ability of the system to perform real-time monitoring and recognition of complex behaviours of people from observing their trajectories within a real, complex indoor environment.*

## 1 Introduction

Surveillance systems for monitoring the behaviour of people have been the focus of much research. Most have worked on human motion recognition in the context of gait recognition, or simple activity detection in limited known spaces. Large area surveillance systems include VSAM [9], which are restricted in the complexity of the behaviours recognised. Thus, one of the challenges is to develop scalable systems for recognition of high level people behaviours in large, complex environments and possibly during extended time intervals.

A review of work in modeling and recognising people's behaviours, especially highly structured behaviours, is presented by Aggarwal and Cai [1]. A simple approach to this problem uses templates, although it is sensitive to variance in different patterns of the same activity, and to noise in the observations. The Finite State Machine (FSM) can be used to model high-level activities [3], but does not account for uncertainty in the model. Alternatively, Hidden Markov Models (HMMs) [19] have been widely used for tackling simple behaviours such as gestures or gait recognition [21, 20]. Other

extensions to the basic HMM have also been used such as the Coupled Hidden Markov Models (CHMMs) for modeling human behaviours and interactions [15], and variable length Markov models (VLMMs) to locally optimise the size of behaviour models [8].

All these approaches employ a flat model of activities. To develop scalable systems for high level behaviour recognition, we need a framework that utilizes the inherent hierarchical structure. Recognizing high-level, semantically rich behaviours has traditionally been the focus of plan recognition work [11, 6]. Sophisticated stochastic models for representing high level behaviours have been used such as Dynamic Bayesian Networks (DBN) [2], Abstract Hidden Markov Models (AHMM) [5], stochastic grammars (including Stochastic Context Free Grammar (SCFG) [17]) and its state-dependent extension Probabilistic State Dependent Grammars (PSDG) [18]. However, most work in this area (with the exception of the AHMM) has been limited to inference at the high level, and the issue of dealing with the low-level noisy sensory models has not been addressed.

Attempts to integrate high level structured behaviour models with low level sensory models have only appeared recently. Oliver [14] proposed a Layered Hidden Markov Model (LHMM) where the classification results of the low layer are used as inputs to the higher layer. Ivanov and Bobick [10] proposed a two-stage strategy: at the low level, the basic HMMs are used to detect simple patterns in the behaviours; at the high level, the outputs produced by the HMMs are interpreted and parsed by a SCFG model of high level behaviours. Alternatively, Nguyen *et al* [13] proposed a fully integrated model for representing both high and low level behaviours based on the Abstract Hidden Markov Model (AHMM).

These approaches have their own shortcomings. Oliver's work separates the task of behaviour classification layer by layer, and the influence of inference in LHMM is only from low level to high level. Ivanov and Bobick's framework is restricted by the context free constraint of the underlying grammar. Complex behaviours, especially goal-directed behaviours, are often "state-dependent", i.e. their evolution depends on the current state of the world which makes them non context-free [16]. Furthermore, the two-stage recognition strategy does not support online, seamless probabilistic inference all the way from low level sensory data to high

level behaviours. Nguyen *et al*'s framework does not have these restrictions, but is limited by the expressive power of the AHMM. Although not context free, the AHMM is entirely state dependent, in the sense that the current behaviour can only be dependent on the current environment state and not on any behaviours that have taken place previously.

In this paper, we present a system architecture for recognition of high-level behaviours of people in large and complex indoor environments. The novelty is in the use of the Abstract Hidden Markov mEmory Model (AHMEM) [4]. This model is as expressive as other grammar-based models, can model state-dependent behaviours, and at the same time support online, scalable and efficient probabilistic inference of high level behaviours from low level data. The hierarchical nature of the model makes it suitable for the natural hierarchy existing in spatial regions, making it scalable to larger and more complex environments.

The paper is organized as follows. An overview of the surveillance system is provided in section 2. The AHMEM framework used for behaviour modelling is described in section 3. Finally, section 4 presents the the experimental results of the implemented system in a real office-type environment.

## 2 Overview of the surveillance system

The surveillance system has two major components: the distributed tracking module and the behaviour recognition module (see Fig. 1). The distributed tracking module extracts people's trajectories using multiple static cameras. The module implements the architecture and tracking algorithms described in [12]. A Central Module (CM) is used to coordinate operations of cameras and maintain the trajectories of people in the scene. There is a Camera Processing Module (CPM) for each camera, which tracks the bounding box of each person in the camera's fields of view using the Kalman filter. The outputs of the Kalman filters are sent to the CM after each time slice where they will be used to form the trajectories of all objects in the scene. The trajectories are passed to the behaviour recognition module to infer behaviours at higher levels.

Since the camera fields of view can overlap, a person in the overlapping area may be viewed from several cameras at a time. In this case, the CM will choose a suitable camera to track the person. Usually, the CM assigns the tracking of a person to the nearest camera to the person. If there is occlusion, the person is assigned to the nearest camera that can offer a clear non-occluded view. Lost objects will be recovered by a matching procedure at the Central Module.

The output of the distributed tracking module is a sequence of coordinates for each object in the scene. Fig. 6 shows examples of typical trajectories returned by the module.

## 3 Behaviour Recognition

The behaviour recognition module takes a sequence of observed coordinates returned by the tracking module and de-

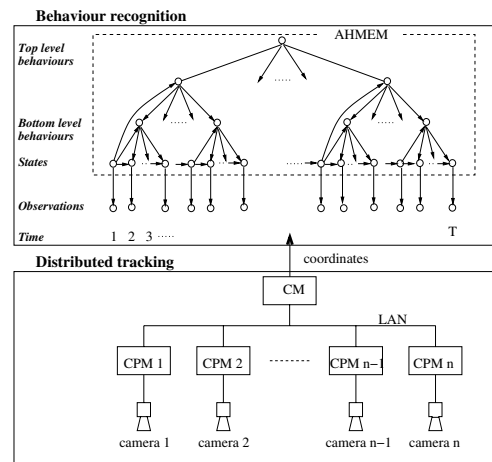


Figure 1. The system architecture.

rites the most probable high level behaviour that matches the observed sequence. For example, if the tracked person approaches a computer, followed by a printer, the system can predict that the most likely activity performed by that person is printing. In reality, the problem is complicated by two factors. Firstly, the observations are noisy due to the camera noise and object occlusion. Secondly, the signature of a single activity may vary, while signatures of different activities may look similar. This means that a simple pattern matching approach for behaviour recognition would perform poorly.

Our solution is based on a Bayesian formulation involving two issues. Firstly, we need a model of how each high-level activity would lead to possible sequences of observations. In other words, we need a model for specifying the conditional distribution  $P(\tilde{o}|\pi)$ , where  $\pi$  denotes an activity and  $\tilde{o}$  denotes a sequence of observations. Secondly, we need an efficient inference procedure to compute  $P(\pi|\tilde{o})$ , the probabilities of different activities given the observed sequence. This would give us a full distribution over the set of possible behaviours. From this, the most likely behaviour can be computed if desired. We employ the Abstract Hidden Markov mEmory Model (AHMEM) [4] as the underlying framework for both the representational and computational tasks.

### 3.1 The Abstract Hidden Markov Memory Model

The Abstract Hidden Markov mEmory Model (AHMEM) [4] is an extension of the AHMM [5], a probabilistic framework for representing and recognising complex behaviours under uncertainty. A behaviour can be refined into a sequence of more simple behaviours at lower levels. The rules for refinement can be made non-deterministic or stochastic if desired. In the language of the AHMM, a behaviour is represented as an *abstract policy*, analogous to a *policy* in Markov Decision Processes (MDP). While a policy in an MDP simply selects an action for execution at each state, an abstract policy is allowed to select other abstract policies in a recur-

sive manner. Each abstract policy  $\pi^*$  has a selection function  $\sigma_{\pi^*} : S_{\pi^*} \times \Pi \rightarrow [0, 1]$  where  $S_{\pi^*}$  is the set of applicable states of  $\pi^*$ ,  $\Pi$  is the set of abstract policies at the lower level which  $\pi^*$  can select from, and  $\sigma_{\pi^*}(s, \pi) = \Pr(\pi | s, \pi^*)$  is the probability that  $\pi^*$  selects the policy  $\pi$  at the state  $s$ . In addition, each abstract policy also has a terminating probability for each state  $\beta_{\pi^*}(s)$ , representing the chance of terminating its execution at state  $s$ . When a hierarchy of such policies is considered, a top-level policy  $\pi^k$  will first select a policy  $\pi^{k-1}$  for execution until  $\pi^{k-1}$  terminates at some state  $s'$ . Then, a new policy  $\pi^{k-1}$  is selected by  $\pi^k$  at state  $s'$  and so on. The policy at the bottom level  $\pi^1$  does not select any other policies, but is modelled as a Markov chain (with termination) within the state space  $S$ . Finally, the state can be made hidden, as in the Hidden Markov Models [19], by considering a set of observations and an observation model  $P(o|s)$ .

The policy hierarchy of the AHMM can be viewed as a stochastic grammar, where the policies correspond to a set of non-terminal symbols, and the selection probabilities for the policies correspond to a set of stochastic production rules. However, in the AHMM framework and its extension, the refinement rule is dependent on the special state variable  $s$  representing the state of the environment. The AHMM language is thus non context-free, and is a type of Probabilistic State Dependent Grammar (PSDG).

Due to its state-dependency, the AHMM has an advantage over the SCFG since the evolution of complex, and especially goal-directed behaviours, depends on the current state of the environment and its relationship to the goal states. The AHMM however is restricted as the way a policy selects a lower-level policy depends only on the current state, and not on any other policies that have been selected in the past. This prevents the AHMM from representing policies that evolve in a number of stages. For example, the behaviour “printing” can be specified in three stages, e.g. “going to computer” followed by “going to printer” followed by “exiting environment”. AHMM can not represent this behaviour since it has no way of remembering the current stage of execution.

The AHMEM [4] removes this restriction by allowing each policy to have internal memory with domain  $M$ . A policy can use its memory variable  $m \in M$  to “remember” its current stage of execution. In the most general form, when a policy  $\pi^*$  commences at some state  $s$ , the memory variable  $m$  can be initialised according to some initial distribution  $\Pr(m|\pi^*, s)$ . Then, each time a lower-level policy terminates and returns at some state  $s'$ , the memory variable for  $\pi^*$  can be updated by the transition probability  $\Pr(m'|m, s', \pi^*)$ . Importantly, the selection and termination model of a policy can be made dependent not only on the current state, but also on the current memory:  $\sigma_{\pi^*}(s, m, \pi) = \Pr(\pi | s, m, \pi^*)$  is the probability that  $\pi^*$  selects  $\pi$  at current state  $s$  when the current memory of  $\pi^*$  is  $m$ ;  $\beta_{\pi^*}(s, m)$  is the probability that  $\pi^*$  terminates at a state  $s$  when the current memory is  $m$ .

The original AHMM is simply a AHMEM with no-memory policies, e.g. when the domain for memory values  $M$  is singleton. A PSDG can also be reduced to a AHMEM

with *linear memory*, i.e. the memory variable in each policy is simply increased by 1 each time it is updated. The AHMEM thus has an expressiveness comparable to grammar-based models. It is also state-dependent, and can handle noisy observation of the state. It is therefore an ideal language for behaviour modelling in our surveillance domain. We now describe an example of using the AHMEM for modelling a behaviour hierarchy in a real office-type environment.

### 3.2 Example of a Behaviour Hierarchy

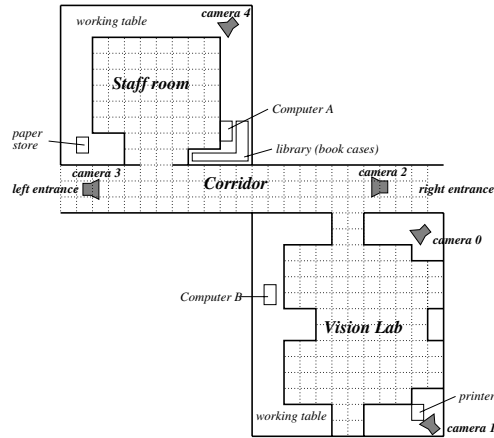


Figure 2. The complex spatial environment.

The environment that we consider consists of three regions: the corridor, the staff room and the vision lab (see Fig. 2 and Fig. 3). People enter and exit the scene via the left or the right entrance. Landmarks represent locations of important objects in the environment: the two computers, the printer, the book cases (library) and the paper store. The environment is modeled by a grid of cells. The cell coordinate of a person’s position is represented by the state variable  $s$ . The observation of a state is the coordinate returned from the distributed tracking component. We assume that the observation

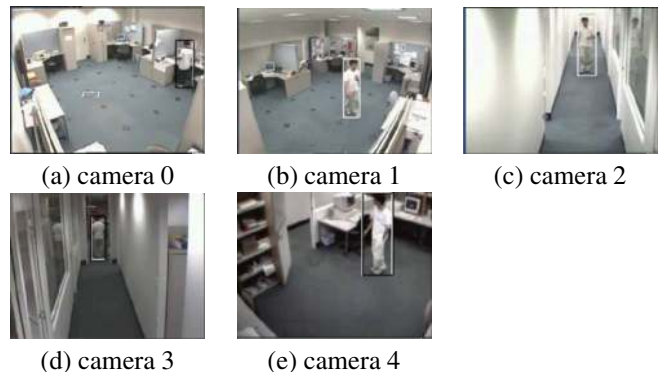


Figure 3. The scene viewed from the five cameras.

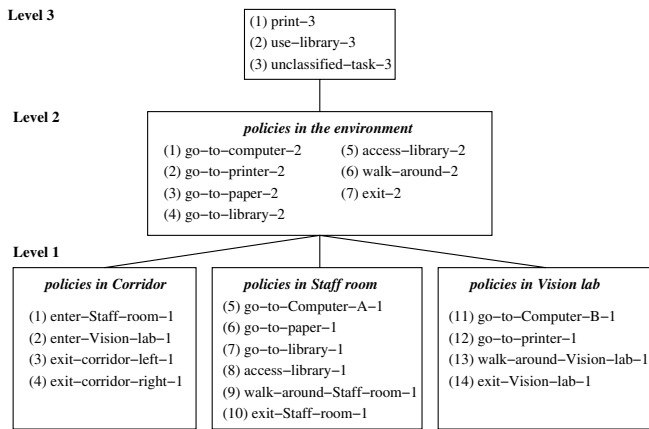


Figure 4. The behaviour hierarchy.

can only be in the 5x5 neighbourhood of the true state. Thus for each camera, the observation model  $\Pr(o|s)$  is defined by a 5x5 matrix specifying the observation likelihood within the neighbourhood of any given state.

Fig. 4 shows a three level behaviour hierarchy defined in this environment. At level 1, the policies represent behaviours that are constrained within a single region. For example,  $go-to-X^1$  where  $X$  is a landmark, is a level 1 policy that takes a person to  $X$ . A level 1 policy is specified by the transition model  $\Pr(s'|s)$ . Together with the camera observation model, these parameters form a HMM. Since from the current state  $s$ , the next state  $s'$  and the observation  $o$  of  $s$  must be in the neighbourhood, the parameters of the HMM are "tied". Therefore, only the conditional probability within a neighbourhood needs to be estimated. We thus learn the parameters  $\Pr(s'|s)$  and  $\Pr(o|s)$  from a set of training video sequences using the expectation maximization (EM) algorithm for HMM with tied parameters.

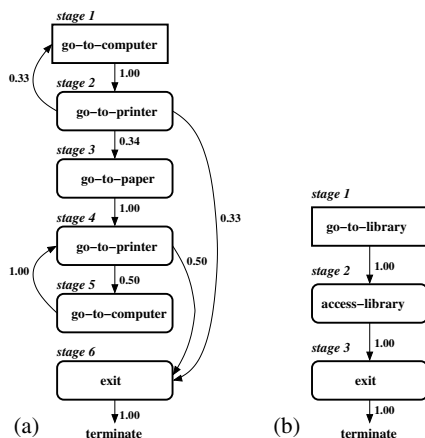


Figure 5. The transition diagrams of the memory variable in behaviours (a)  $print^3$  and (b)  $use-library^3$ .

The level 2 policies represent the movement of a person within the entire environment, such as going to a particular landmark from a position anywhere in the environment. For example,  $go-to-computer^2$  is a policy that takes the user to the nearest computer. This is modelled as a level 2 policy  $\pi^2$  in the AHMEM which selects a level 1 policy for execution depending on the current state  $s$  (representing the current position of the person). If  $s$  is inside the staff room, the person is most likely to go to *Computer-A*, thus the conditional distribution  $\Pr(\pi^1|s, \pi^2)$  would peak at the value  $\pi^1 = go-to-Computer-A^1$ . Similarly, if the person is in the vision lab, the person is most likely to use *Computer-B*, and the conditional distribution  $\Pr(\pi^1|s, \pi^2)$  would peak at the value  $\pi^1 = go-to-Computer-B^1$ . Note that a level 2 policy defined this way is state-dependent, but memoryless, since the selection probabilities do not depend on the value of the memory variable  $m$ .

The level 3 policies represent the different tasks that a person might perform during the entire interval that the person stays in the environment, e.g. printing document, accessing the library, or simply walking around. For example,  $print^3$  is represented by a level 3 policy  $\pi^3$  in the AHMEM. Unlike the memoryless policies defined at level 2, this policy has an internal memory variable whose transition diagram is shown in Fig. 5(a). The printing behaviour thus involves first going to a computer followed by going to the printer. If however the printer has run out of paper, the person has to go and fetch more paper, then come back to the printer. Note that the memory transition probability  $P(m'|m, s, \pi^3)$  does not depend on the current state  $s$ . In addition, the memory variable in Fig. 5(a) immediately determines which policy at level 2 should be selected. For example,  $\Pr(\pi^2|s, m = go-to-paper, \pi^3) = 1$  when  $\pi^2 = go-to-paper^2$ . Therefore, the definition of a third level policy  $\pi^3$  has memory but is state-independent, and thus is similar to a set of SCFG production rules. However, the entire policy hierarchy is not context-free due to the state-dependency at level 2.

Since the parameters at level 2 and 3 are intuitive, their values are manually chosen as in Fig. 5. However, they can be easily learned from labelled training data by estimating the frequency of the next landmark given the current landmark.

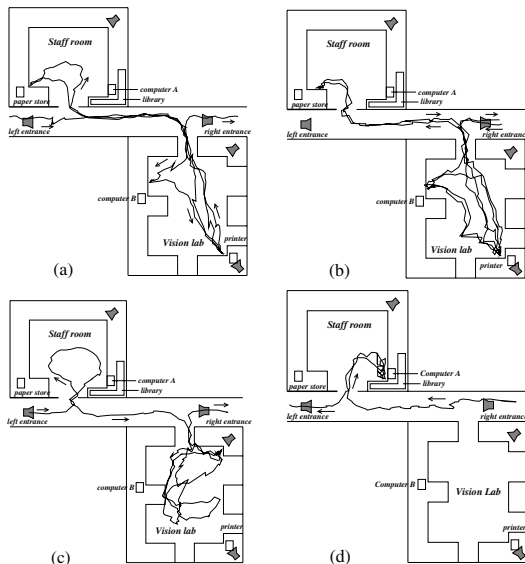
### 3.3 Inferring Behaviours from Observations

The AHMEM and its parameters define a conditional distribution over the observation sequences given a policy:  $\Pr(\tilde{o}|\pi^k)$ . In recognising the behaviour of a person in the scene, we are given a sequence of observations  $\tilde{o}_{t-1} = (o_1, \dots, o_{t-1})$  up to the current time  $t$ , and need to compute the probability  $\Pr(\pi_t^k|\tilde{o}_{t-1})$ , where  $\pi_t^k$  represents the policy being executed at level  $k$  and time  $t$ . This provides the distribution of the possible behaviours that might be currently executed at level  $k$  in the hierarchy. The computation needs to be done at every time instance  $t$  when a new observation  $o_t$  arrives. The problem is termed *policy recognition* [5], and is equivalent to the on-line inference (filtering) problem in the AHMEM. Exact solutions to this problem will have exponen-

tial complexity in the number of levels of the policy hierarchy, and thus can not scale up to domains of larger sizes.

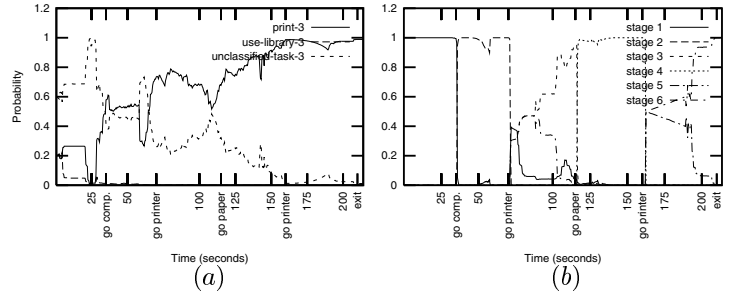
Fortunately, there is an efficient approximate inference algorithm based on the Rao-Blackwellised Particle Filter (RBPF) [7] for computing these probabilities [4, 5]. The RBPF algorithm gains its efficiency by exploiting the special probabilistic independence structure of the AHMEM, due to the special mechanism for selecting policies at the lower levels. The inference algorithm also supports other queries involving the current variables in the AHMEM. For example, assuming that the current policy at level  $k$  is  $\pi_t^k$ , then its current stage of execution can be determined from the probability  $\Pr(m_t^k | \pi_t^k, \delta_{t-1})$  where  $m_t^k$  is the current memory variable at level  $k$ . The current state (position) of the person can also be determined by computing  $\Pr(s_t | \delta_{t-1})$ . If AHMEM inference engine gets no observation due to temporary occlusion in the scene, the engine still carries on the inference step by rolling forward the belief state, but not updating it with any observation.

#### 4 Experiments and Results



**Figure 6.** (a) Person A's trajectory in scenario 1: goes to Computer B, to the printer, to the paper store, to the printer again and exits. (b) Person B's trajectory in scenario 2: goes to Computer B, to the printer, to the Computer B again, to the printer again, to the paper store, to the printer for the third time and exits. (c) Person C's trajectory in scenario 2: walks around in the Staff room, walks around in the Vision lab and exits. (d) Person C's trajectory in scenario 3: goes to the library, uses the library and exits.

The AHMEM and its RBPF inference algorithm have been



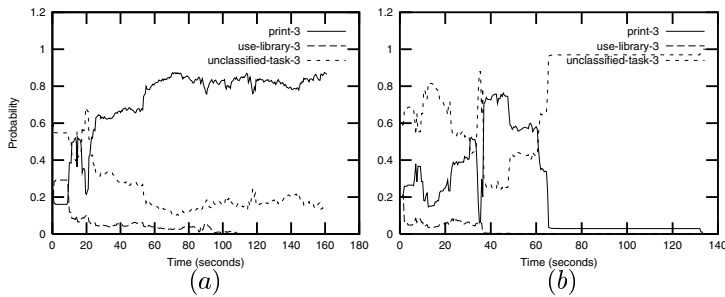
**Figure 7.** (a) Querying the top level behaviours of person A in scenario 1. (b) The progress of executing behaviour  $print^3$  of person A in scenario 1.

used for implementing the behaviour recognition module in our surveillance system. The surveillance system as shown in Fig. 2 has five static cameras. Fig. 3 shows the scene viewed from these cameras. The behaviour hierarchy is as outlined in section 3.2.

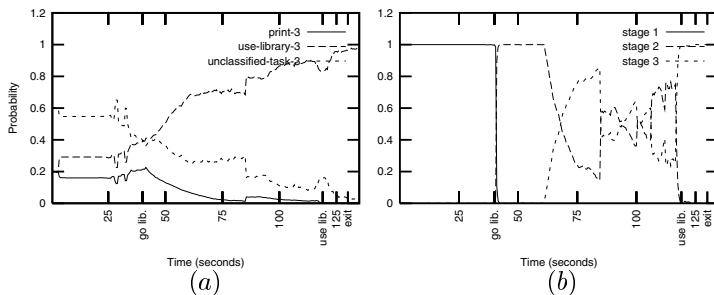
We examine how the system tracks the top level behaviours (level 3) which are  $print^3$ ,  $use-library^3$  and  $unclassified-task^3$ . We consider three scenarios. In scenario 1, person A enters the scene from the left entrance, visits the Staff room and the Vision lab, then goes out of the scene via the right entrance. The trajectory of person A detected by the multiple cameras is shown in Fig. 6(a). Fig. 7(a) shows the probabilities that person A is executing behaviours  $print^3$ ,  $use-library^3$  or  $unclassified-task^3$ . These probabilities are denoted by  $p_{print-3}$ ,  $p_{lib-3}$  and  $p_{unc-3}$  respectively. At the beginning  $p_{unc-3}$  is the highest because the  $unclassified-task^3$  has the highest prior. Then  $p_{print-3}$  starts to increase when person A approaches Computer B. When he leaves computer B and approaches the printer,  $p_{print-3}$  becomes higher than  $p_{unc-3}$  and remains high while person A goes to the paper store, returns to the printer and exits the environment. Fig. 7(b) shows the probability of the current memory of behaviour  $print^3$  over time. It can be seen that person A executes stages 1, 2, 3, 4 and 6 consecutively. This means that the sequence of level 2 behaviours executed by person A is  $go-to-computer^2$ ,  $go-to-printer^2$ ,  $go-to-paper^2$ ,  $go-to-printer^2$  and  $exit^2$ .

In scenario 2, there are two people B and C in the scene. The trajectories of the two people are shown in Fig. 6(b) and Fig. 6(c). Fig. 8(a) shows the results of querying the top level behaviours of person B. As in the figure, the system recognises  $print^3$  as the winning behaviour of person B. This is consistent with the fact that person B goes to computer B, to the printer, to computer B again, to the printer again, to the paper store, to the printer for the third time and exits. The system also recognises correctly  $unclassified-task^3$  as the winning behaviour of person C (see Fig. 8(b)).

Fig. 6(c) shows the trajectory of person D in scenario 3. The system correctly finds that  $use-library^3$  is the winning top level behaviour (see Fig. 9(a)). The results of querying the stage transition of behaviour  $use-library^3$  are shown in Fig.



**Figure 8.** Querying the top level behaviours of (a) person B and (b) person C in scenario 2.



**Figure 9.** (a) Querying the top level behaviours of person D in scenario 3. (b) The progress of executing behaviour *use-library*<sup>3</sup> of person D in scenario 3.

9(b) and this corresponds to the 3 stages of this behaviour shown in Fig. 5(b).

The results in the three scenarios are obtained by using RBPF algorithm with 3000 samples. The average processing time for each observation is approximately 0.9 sec on an AMD Athlon(TM) XP1700+ machine. The results show that the system is able to correctly recognise the activities being modelled, and monitor the progress of these activities in real time.

## 5 Conclusion

We have developed a surveillance system for recognising and monitoring high-level human behaviours from multi-camera surveillance data. Using the AHMEM as the underlying framework, the system can query the high-level behaviours executed by a person over time and detect the period of time which the person executes each sub-behaviour. Preliminary results demonstrate the ability of the system to provide real-time monitoring of high level behaviours in complex spatial environments.

## References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

[2] D. Albrecht, I. Zukerman, and A. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modelling and User-adapted Interaction*, 8(1–2):5–47, 1998.

[3] D. Ayers and M. Shah. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing*, 19(12):833–846, October 2001.

[4] H. H. Bui. Efficient approximate inference for online probabilistic plan recognition. *AAAI Fall Symposium on Intent Inference for Users, Teams and Adversaries*, 2002.

[5] H. H. Bui, S. Venkatesh, and G. West. Policy recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

[6] E. Charniak and R. P. Goldman. A Bayesian model of plan recognition. *Artificial Intelligence*, pages 53–79, 1993.

[7] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence*. AAAI Press, 2000.

[8] A. Galata, N. Johnson, and D. Hogg. Learning variable length Markov models of behaviour. *Int. Journal of Comp. Vision and Image Understanding*, 81(3):398–413, 2001.

[9] E. Grimson. VSAM. URL: <http://www.ai.mit.edu/projects/darpa/vsam>, 1998.

[10] Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):852–872, August 2000.

[11] H. Kautz and J. F. Allen. Generalized plan recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 32–38, 1986.

[12] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Coordination of multiple cameras to track multiple people. In *Asian Conference on Computer Vision*, pages 302–307, 2002.

[13] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Hierarchical monitoring of people’s behaviors in complex environments using multiple cameras. In *International Conference on Pattern Recognition*, Quebec City, QC, August 2002.

[14] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Fourth IEEE Int. Conf. on Multimodal Interfaces*, pages 3–8, 2002.

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

[16] D. Pynadath. *Probabilistic Grammars for Plan Recognition*. PhD thesis, University of Michigan, 1999.

[17] D. V. Pynadath and M. P. Wellman. Generalized queries on probabilistic context-free grammars. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):65–77, 1998.

[18] D. V. Pynadath and M. P. Wellman. Probabilistic state-dependent grammars for plan recognition. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 507–514, San Francisco, CA, 2000.

[19] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[20] T. Starner and A. Pentland. Visual recognition of american sign language using Hidden Markov Models. In *International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[21] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 379–385, 1992.