

Discovery of Activity Structures Using the Hierarchical Hidden Markov Model

Nam Nguyen, Svetha Venkatesh
Department of Computing, Curtin University of Technology
GPO Box U1987 Perth, 6845 Western Australia
{nguyentn,svetha}@cs.curtin.edu.au

Abstract

In building a surveillance system for monitoring people behaviours, it is important to understand the typical patterns of people's movement in the environment. This task is difficult when dealing with high-level behaviours. The flat model such as the hidden Markov model (HMM) is inefficient in differentiating between signatures of such behaviours. This paper examines structure learning for high-level behaviours using the hierarchical hidden Markov model (HHMM). We propose a two-phase learning algorithm in which the parameters of the behaviours at low levels are estimated first and then the structures and parameters of the behaviours at high levels are learned from multi-camera training data. Our algorithm is then evaluated using data from a real environment, demonstrating the robustness of the learned structure in recognising people's behaviour.

1 Introduction

Building automated surveillance systems using multiple cameras has been the focus of much research [1, 5, 9, 10, 11]. In these systems, the crucial task is detecting and monitoring human behaviours. Due to noise from cameras and the complexity of people's behaviours, especially high-level behaviours, this task is difficult and requires sophisticated methods. We consider high level behaviours that can be refined. For example, the high-level behaviour *print* can be constructed from three low-level behaviours: (1) *go_to_computer*, (2) *go_to_printer*, and (3) *go_to_paper_store*. A person executing the behaviour *print* can first *go_to_computer*, then *go_to_printer*, and then *go_to_computer* again to issue new print commands. If the printer has run out of paper, the person has to *go_to_paper_store*, then *go_to_printer* to fetch more paper. Understanding the structure of the high-level behaviours is important for recognising both high-level and low-level people's behaviours reliably. However, the structures of the high-level behaviours are not easy to specify manually. Efficient approaches for tackling this problem require discovering the behaviour structure automatically from the training data.

The hidden Markov model (HMM) has been used in much research for learning and recognising simple behaviours [4, 7]. However, this flat model is inefficient for modelling high-level behaviours because they cannot characterise the hierarchical nature inherent in

behaviours. Recent approaches using hierarchical probabilistic models such as the layered hidden Markov model (LHMM) [13], stochastic context free grammar (SCFG) [15], the abstract hidden Markov model (AHMM) [3] or the hierarchical hidden Markov model (HHMM) [6, 2] are robust in modelling and recognising high-level behaviours. Osentoski *et al.* [14] present a behaviour tracking system, in which the behaviours are modelled by the AHMM [3]. The model’s parameters are learned from the training data using the expectation maximisation (EM) algorithm. But the special landmarks in the environment are not be exploited for defining a richer class of behaviours. Liao *et al.* [10] introduce a surveillance system to learn people’s daily behaviours using GPS sensors. The AHMM is used in the system for representing the behaviours. First, the EM algorithm is used to learn the people’s goals and important locations. The remaining parameters are then estimated using the Monte Carlo EM [17] method. The work of Osentoski *et al.* and Liao *et al.* does not explicitly define the structure of the high-level behaviour. The system proposed by Nguyen *et al.* [11] does not have this limitation because of the use of the abstract hidden Markov memory model (AHMEM) – an extension of the AHMM – for modelling the behaviour hierarchy. However, the behaviour structures in this system are specified manually by observing the typical movements of people executing each specific high-level behaviour, and not automatically learned from the training data.

This paper aims to use the HHMM [6, 2] for tackling two issues: (1) modelling high-level behaviours in indoor environments and (2) discovering the structure of high-level behaviours from the training data given multiple cameras. Each high-level behaviour is initialised as a fully connected structure of all behaviours at the lower level (Fig. 1(a)). The task is to prune this fully connected structure using the training data (Fig. 1(b)) and to learn the probability of the structural links. We argue that the use of the HHMM for structure discovery of high-level behaviours allows us to recognise both the high-level and low-level behaviours more reliably. The novelty of this paper lies in the introduction of a two-phase learning algorithm to discover the behaviour structures and to estimate the probabilities for the links within these structures. We evaluate the two-phase learning algorithm using real data.

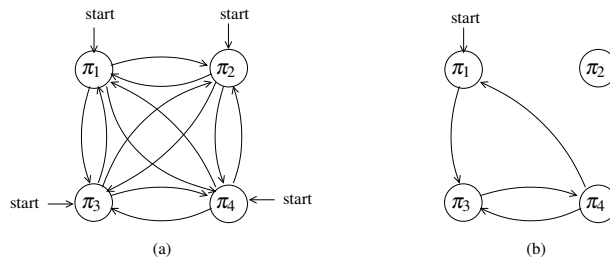


Figure 1: (a) The initial structure of a high-level behaviour. All behaviours at the lower level (π_1 , π_2 , π_3 , and π_4) are fully connected. (b) The structure of the high-level behaviour learned from the training data where some links are pruned from the initial structure.

The layout of the paper is as follows. Section 2 presents the techniques used in modelling the behaviour hierarchy for the system. Section 3 describes the algorithm for learning the structures of high-level behaviours. The experimental results in a real environment are given in Section 4, followed by our concluding remarks in Section 5.

2 The Behaviour Hierarchical Model

2.1 The environment and behaviours

The environment is divided into a set of discrete states. Some landmarks are specified in the environment. An example of the environment, which is a room, is shown in Fig. 2(a). The room is monitored by two static cameras. The special landmarks in the room are: the door, TV chair, fridge, coffee machine, stove, cupboard, and dining table. Note that the coffee machine and the fridge are at the same location. There are 24 states in the room, which are numbered 1, 2, . . . , 24. The size of each state is about $1\text{m} \times 1\text{m}$. The x-axis, y-axis, original coordinates, and the states of the environment are shown in Fig. 2(b).

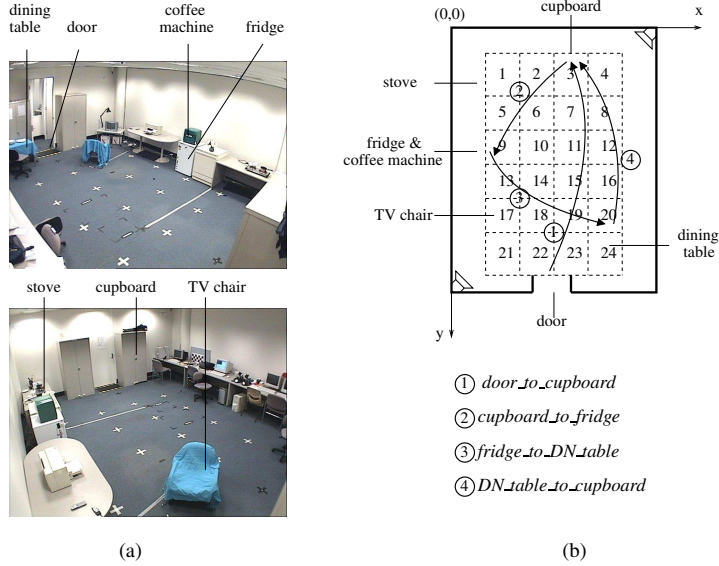


Figure 2: (a) The room viewed from two static cameras. (b) The states in the room and primitive behaviours 1, 2, 3 and 4.

The system aims to recognise two classes of behaviours: *primitive behaviours* and *complex behaviours*. The primitive behaviour represents a person's action of going from a specific landmark in the environment to another. Fig. 2(b) shows examples of four primitive behaviours: (1) *door_to_cupboard*, (2) *cupboard_to_fridge*, (3) *fridge_to_DN_table*, and (4) *DN_table_to_cupboard*. The complex behaviour is defined from a set of primitive behaviours and can be refined into different sequences of the primitive behaviours. For example, the complex behaviour *have_coffee* can be a sequence of primitive behaviours (1), (2), (3), and (4).

2.2 Parameters for the primitive behaviour

Each primitive behaviour π^1 has a starting landmark and an ending landmark. It is specified by the following parameters:

- (1) The initial state probability $I^{\pi^1}(s)$: the probability that π^1 starts from the state s .

$I^{\pi^1}(s)$ is assigned a high value if s is near the starting landmark of π^1 ; otherwise $I^{\pi^1}(s)$ is assigned a small value.

(2) The movement model $A^{\pi^1}(s, s')$: the probability that the person moves from the current state s to the next state s' . We assume that a person can only move from the current state to one of the neighbouring states after each time slice. Thus, $A^{\pi^1}(s, s') = 0$ if s' is not a neighbouring state of s .

(3) The ending state probability $E^{\pi^1}(s)$: the probability that π^1 terminates at the state s . $E^{\pi^1}(s)$ is high if s is near the ending landmark; otherwise $E^{\pi^1}(s)$ is low.

2.3 Parameters for the complex behaviour

Each complex behaviour π^2 is defined from a set of primitive behaviours. The structure of π^2 can be represented by a diagram in which the nodes represent the primitive behaviours. The complex behaviour π^2 has the following parameters:

(1) $I^{\pi^2}(\pi^1)$: the probability that π^2 selects the primitive behaviour π^1 to start its execution.

(2) $A^{\pi^2}(\pi^1, \pi^{1'})$: the probability that π^2 selects the next primitive behaviour $\pi^{1'}$ to execute when the current primitive behaviour π^1 terminates.

(3) $E^{\pi^2}(\pi^1)$: the probability that π^2 terminates when the current primitive behaviour π^1 terminates.

The complex behaviour π^2 is executed as follows. First, π^2 selects a primitive behaviour π^1 from the distribution $I^{\pi^2}(\pi^1)$ to start its execution. Then, π^1 is executed until it terminates. At this time, π^2 can terminate with probability $E^{\pi^2}(\pi^1)$. If π^2 does not terminate, it continues to select another primitive behaviour $\pi^{1'}$ from the distribution $A^{\pi^2}(\pi^1, \pi^{1'})$ for execution. The loop continues until π^2 terminates.

3 Discovering the Structure of the Complex Behaviours

Usually, the structure of a complex behaviour π^2 is unknown. The important question is to learn the structure of π^2 from the training data obtained from multiple cameras. To address this issue, we use the HHMM, which is an extension of the well-known HMM. The HHMM with its inherent hierarchical structure is suitable for representing the behaviour hierarchy and dealing with noise from multi-camera data. Moreover, efficient learning algorithms exist to estimate the model parameters from a set of observation sequences. For this study, the behaviour hierarchy is first mapped to an HHMM, and then the EM algorithm is used to learn the behaviours.

3.1 The hierarchical hidden Markov model

The hierarchical hidden Markov model (HHMM) [6, 2] is an extension of the HMM [16] to include a hierarchy of hidden states. A special *end* state is introduced at each level to signal when control of the activation is returned to the state at the higher level. An HHMM has the following components: a topological structure ζ , an observation alphabet \mathcal{S} and a set of parameters θ . The topology ζ specifies the depth of the model, the state space at each level, and the parent-children relationship between two consecutive levels. The states at the lowest level are called *production* states. The states at higher

levels are called *abstract* states. Only production states can generate observations. The set of parameters θ consists of the observation model $B_{y|p}$, the initial distribution I^{d,p^*} , the transition probability $A_{i,j}^{d,p^*}$ and the ending distribution $A_{i,\text{end}}^{d,p^*}$, where $y \in \mathcal{Y}$ is an observation, p is a production state, p^* is an abstract state at level d , and i, j are children of p^* . The parameter $B_{y|p}$ is the probability of observing y given that the production state is p . I^{d,p^*} is the initial distribution over the set of the children of p^* . $A_{i,j}^{d,p^*}$ is the transition probability from child i to child j . $A_{i,\text{end}}^{d,p^*}$ is the probability that p^* terminates given its current child i terminates. A representation of the HHMM as a dynamic Bayesian network (DBN) is provided in [2], which defines a joint probability distribution (JPD) over the set of all variables $\{p_t^d, e_t^d, y_t \mid \forall(t, d)\}$, where p_t^d is the state at level d and time t , e_t^d represents whether p_t^d terminates or not, and y_t is the observation at time t .

3.2 Mapping the behaviour model to the HHMM

We use the HHMM to model the primitive and complex behaviours of a person over time. Fig. 3 shows the DBN representation of the model at two consecutive time slices. The nodes π_t^1 and π_t^2 represent the primitive and complex behaviours at time t , respectively. The nodes e_t^1 and e_t^2 represent the end status of the primitive and complex behaviours, respectively. e_t^1 and e_t^2 can be either *True* or *False*. Note that, the complex behaviour cannot terminate when the primitive behaviour is still continuing, thus $e_t^1 = \text{False} \Rightarrow e_t^2 = \text{False}$. The state s_t represents the current position of the person. The observation of s_t is $o_t = (x_t, y_t)$, where (x_t, y_t) are the person's coordinates that are returned from the tracking system of multiple cameras. We use a mixture of Gaussians for the observation model, where the mixture variable c_t represents the camera generating the observation o_t . The conditional probabilities for the links in the HHMM are obtained from the parameters of the primitive and complex behaviours as follows:

$$\begin{aligned}
\Pr(e_t^2 | \pi_t^2, \pi_t^1, e_t^1) &= \begin{cases} E\pi_t^2(\pi_t^1) & \text{if } e_t^1 = \text{True} \\ \delta_{e_t^2, \text{False}} & \text{otherwise} \end{cases} \\
\Pr(e_t^1 | \pi_t^1, s_t) &= E\pi_t^1(s_t) \\
\Pr(\pi_{t+1}^2 | \pi_t^2, e_t^2) &= \begin{cases} \delta_{\pi_{t+1}^2, \pi_t^2} & \text{if } e_t^2 = \text{False} \\ I(\pi_{t+1}^2) & \text{otherwise} \end{cases} \\
\Pr(\pi_{t+1}^1 | \pi_{t+1}^2, \pi_t^1, e_t^2, e_t^1) &= \begin{cases} \delta_{\pi_{t+1}^1, \pi_t^1} & \text{if } e_t^1 = \text{False} \\ A^{\pi_{t+1}^2}(\pi_t^1, \pi_{t+1}^1) & \text{if } e_t^1 = \text{True}, e_t^2 = \text{False} \\ I^{\pi_{t+1}^2}(\pi_{t+1}^1) & \text{if } e_t^1 = \text{True}, e_t^2 = \text{True} \end{cases} \\
\Pr(s_{t+1} | \pi_{t+1}^1, s_t, e_t^1) &= \begin{cases} A^{\pi_{t+1}^1}(s_t, s_{t+1}) & \text{if } e_t^1 = \text{False} \\ I^{\pi_{t+1}^1}(s_{t+1}) & \text{otherwise} \end{cases}
\end{aligned}$$

where $\delta_{i,j}$ the Kronecker delta symbol, defined by $\delta_{i,j} = 1$ if $i = j$, otherwise $\delta_{i,j} = 0$; $I(\pi_{t+1}^2)$ is the prior distribution of the complex behaviours.

3.3 Learning the structure of complex behaviour

The structure of a complex behaviour π^2 is initialised such that it is fully connected to all primitive behaviours. The probabilities for the links are initialised uniformly. We aim

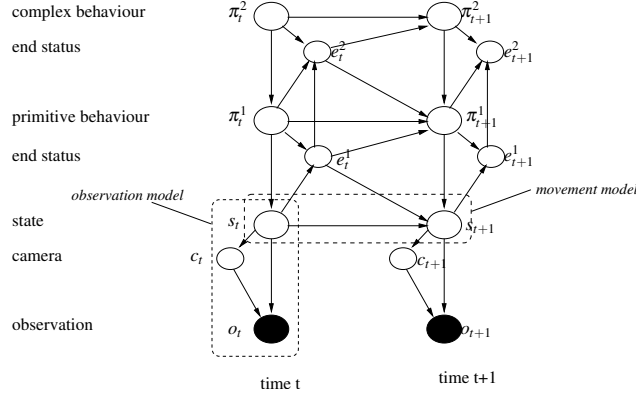


Figure 3: Two time slices of the DBN representation of the behaviour model.

to learn the probabilities for the links from the training data, then prune the links that are assigned too small probabilities. To do this task, we need to estimate the parameters of π^2 – that is, I^{π^2} , A^{π^2} , and E^{π^2} . Assuming that the complex behaviour does not terminate during our examining time, we can ignore the ending probability E^{π^2} . The remaining parameters (I^{π^2} and A^{π^2}) can be obtained from the parameters θ of the HHMM. Thus, the task of discovering the complex behaviour structures can be done by learning the parameters θ of the HHMM. First, the observation models – that is, $\Pr(o|s, c)$ and $\Pr(c|s)$ (Fig. 3) – are estimated by manually extracting the true position and the camera corresponding to observations. Then, the remaining parameters of θ are learned from the training data using one-phase or two-phase learning algorithms. Note that the training data is a set of observation sequences $\mathcal{O} = \{\mathcal{O}^1, \mathcal{O}^2, \dots, \mathcal{O}^K\}$, where \mathcal{O}^k ($1 \leq k \leq K$) is the trajectory of a person executing a complex behaviour in the environment: $\mathcal{O}^k = (x_1, y_1), (x_2, y_2), \dots, (x_{T_k}, y_{T_k})$.

One-phase learning algorithm. The parameters θ of the HHMM are estimated from the training data in one integrated step using the junction tree algorithm [8] or the Asymmetric Inside-Outside (AIO) [2]. This method requires a large amount of training data to estimate θ correctly.

Two-phase learning algorithm. The algorithm has two phases: (1) learning the movement model for each primitive behaviour π^1 – that is, $\Pr(s'|\pi^1, s)$ – and (2) learning the remaining parameters of θ from the training data. The details of each phase are outlined below:

The movement model of π^1 is learned in a similar manner to [12]. The movement model of π^1 and the observation models form a hidden Markov model with mixture of Gaussians (Fig. 3). The movement model is compressed into a 3×3 matrix specifying the probability that a person executing π^1 moves to the neighbouring states (including the current state). Given a direction to reach the ending landmark of π^1 , the matrix can be rotated to apply the probabilities. We use the EM algorithm with the compressed parameters [12] to learn the movement model from a set of training observation sequences of π^1 . The idea of this algorithm is that, after each iteration, the derived movement model is compressed into a 3×3 matrix, and then is expanded to the full movement model before being used in the next iteration.

After obtaining the observation models and movement model, we use the EM algorithm to learn the remaining parameters for the HHMM. The EM has two steps: the Expectation step (E-step) and the Maximisation step (M-step). The E-step calculates the expected sufficient statistic (ESS) for θ . Methods used to calculate the ESS include the junction tree algorithm [8], the AIO algorithm [2] or the Rao-Blackwellise particle filter (RBPF) [3]. For the three-level HHMM in our system, an exact method such as the junction tree algorithm is most suitable. In the M-step, the result of calculating the ESS for θ is normalised to obtain a new value for θ .

We derived the parameters I^{π^2} and A^{π^2} of the complex behaviour π^2 from the parameters θ of the HHMM. Then, the structure of π^2 is created from I^{π^2} and A^{π^2} after removing the elements that are assigned too small values.

4 Experimental Results

4.1 Implementation

The system is implemented in the environment in Fig. 2(a) and (b). The set of primitive behaviours are:

- | | | |
|---------------------------------|---------------------------------|----------------------------------|
| (1) <i>door_to_cupboard</i> | (5) <i>door_to_TV_chair</i> | (9) <i>stove_to_fridge</i> |
| (2) <i>cupboard_to_fridge</i> | (6) <i>TV_chair_to_cupboard</i> | (10) <i>DN_table_to_stove</i> |
| (3) <i>fridge_to_DN_table</i> | (7) <i>fridge_to_TV_chair</i> | (11) <i>stove_to_DN_table</i> |
| (4) <i>DN_table_to_cupboard</i> | (8) <i>door_to_stove</i> | (12) <i>cupboard_to_TV_chair</i> |

The set of complex behaviours are *have_coffee*, *have_snack*, and *have_meal*. Each complex behaviour is initialised as a fully connected structure of the 12 primitive behaviours. The probability of that a complex behaviour selects a primitive behaviour to start its execution and the probability on the links of the behaviour structures are initialised uniformly.

4.2 Learning observation model and movement model

We collect a set of 1127 tuples (position, camera, observation) to learn the observation models – that is, $\Pr(o|s, c)$ and $\Pr(c|s)$. The observations are obtained from the multiple camera tracking system. The person’s position and camera corresponding to an observation are obtained by manually analysing the video data. The observation model for the state $s = 1$, which is calculated from the set of tuples (position, camera, observation), is:

$$\begin{aligned} \Pr(o|s = 1, c = \text{Camera 1}) &= N(o; (1.35, 1.55), \begin{bmatrix} 0.016 & 0 \\ 0 & 0.031 \end{bmatrix}) \\ \Pr(o|s = 1, c = \text{Camera 2}) &= N(o; (1.55, 1.44), \begin{bmatrix} 0.046 & 0 \\ 0 & 0.045 \end{bmatrix}) \\ \Pr(c = \text{Camera 1}|s = 1) &= 0.47 \\ \Pr(c = \text{Camera 2}|s = 1) &= 0.53 \end{aligned}$$

For each primitive behaviour, we collect 15 training observation sequences corresponding to that behaviour. The movement model of the primitive behaviour is learned from these training sequences. Fig. 4 shows the movement models of behaviours *door_to_cupboard* and *door_to_stove* learned from the training sequences.

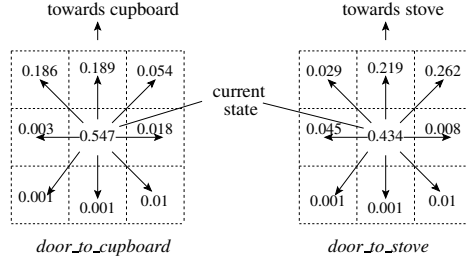


Figure 4: The movement models of primitive behaviours *door_to_cupboard* and *door_to_stove* learned from the training sequences.

4.3 Learning the structure of complex behaviours

The training data to learn the structure of the complex behaviours *have_coffee*, *have_snack* and *have_meal* are obtained from 45 scenarios. In each scenario, a person executes one of the three complex behaviours. The person’s trajectory is obtained from the tracking system. As a result, we have 45 observation sequences. Note that the label of each sequence – that is, the name of the corresponding complex behaviour – is available for the learning process.

The structures of the complex behaviours is learned from the training data using the two-phase algorithm in Section 3.3. Table 1 shows the results of learning the complex behaviour *have_coffee*. The columns and rows corresponding to primitive behaviours 5, 6, 8, 9, 10, 11, and 12 are not shown in the table because the complex behaviour *have_coffee* always starts by executing primitive behaviour 1 and the probabilities that primitive behaviours 5, 6, 8, 9, 10, 11, or 12 are executed by behaviour *have_coffee* are insignificant.

We prune the elements of I^{π^2} and A^{π^2} that are assigned too small values to derive the structure of behaviours. Fig. 5(a) shows the structure of the complex behaviour *have_coffee*. Note that primitive behaviour 3 (*fridge_to_DN_table*) can be executed after primitive behaviour 7 (*fridge_to_TV_chair*) terminates because the TV chair is near the fridge. Fig. 5(b) and (c) show the structure of the complex behaviours *have_snack* and *have_meal*, respectively.

Primitive behaviours	$I^{\pi^2}(\pi^1)$	$A^{\pi^2}(\pi^1, \pi^{1'})$				
		$\pi^{1'} = 1$	2	3	4	7
$\pi^1 = 1$	1.00	0	1.00	0	0	0
2	0	0	0	0.96	0	0.04
3	0	0	0	0	1.00	0
4	0	0	1.00	0	0	0
7	0	0	0	0.99	0	0

Table 1: The parameters of behaviour $\pi^2 = \textit{have_coffee}$ learned from the 45 labeled observation sequences.

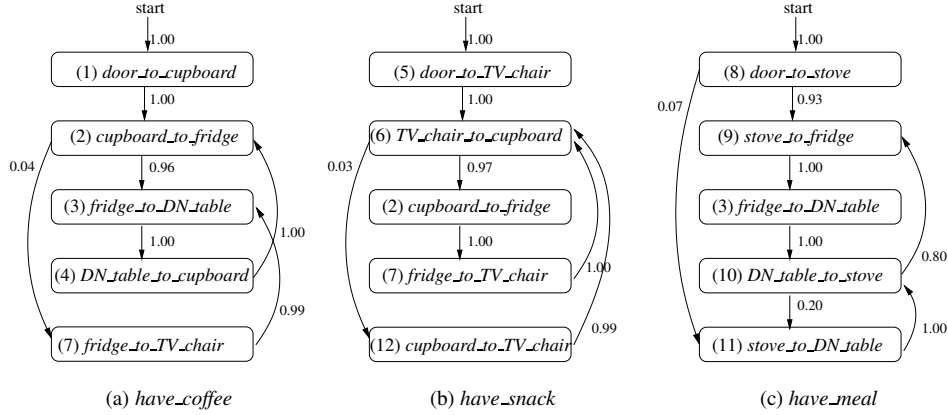


Figure 5: The structures of the complex behaviours (a) *have_coffee*, (b) *have_snack*, and (c) *have_meal* learned from the 45 labeled observation sequences.

4.4 Evaluating the performance of the learned complex behaviours

We test the system to recognise the behaviours in 30 test sequences using the learned complex behaviours. From the recognition results, we compute the *accuracy rate* and *correct duration* that are defined as follows. First, the *winning complex behaviour* of an observation sequence is the complex behaviour that is assigned the highest probability at the end of the sequence. Then, the *accuracy rate* is the ratio of the number of observation sequences, of which the *winning complex behaviour* matches the ground truth, to the total number of test sequences. The *correct duration* is defined as the total of the time period, in which the primitive behaviour assigned the highest probability matches the ground truth, to the length of the observation sequence. The *accuracy rate* and *correct duration* refer to the recognition performance at the complex behaviour level and the primitive behaviour level, respectively.

We compare the reliability of the two-phase algorithm with the one-phase algorithm in learning the structures of the complex behaviours. We use the complex behaviours learned by the two algorithms to recognise the behaviours in the 30 test sequences. The test results are shown in Table 2. The *accuracy rate* of the model learned by the two-phase algorithm is equal to that of the model learned by the one-phase algorithm. However, the *correct duration* when using the two-phase algorithm is much higher than the *correct duration* when using the one-phase algorithm (86.5% versus 21.3%). This means that the two-phase algorithm is more efficient than the one-phase algorithm in discovering the structure of the complex behaviours.

5 Conclusion

We have presented the techniques of using the HHMM for representing the behaviour hierarchical model. We propose the use of the two-phase algorithm for discovery of the structures of complex behaviours. Given a set of primitive behaviours and complex behaviours, the system is able to learn the behaviour structures and compute the appropriate parameters from multi-camera training data. The experimental results in a real environ-

	Accuracy rate	Correct duration
Model learned by the two-phase algorithm	100%	86.5%
Model learned by the one-phase algorithm	100%	21.3%

Table 2: The *accuracy rate* and *correct duration* in recognising the people’s behaviours with different models.

ment demonstrate that the use of the model learned by the two-phase algorithm outperforms the one-phase algorithm in behaviour recognition.

References

- [1] 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.
- [2] H. Bui, D. Phung, and S. Venkatesh. Hierarchical hidden Markov models with general state hierarchy. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 324–329, San Jose, California, 2004.
- [3] H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden Markov model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.
- [4] G. Cielniak, M. Benezit, and W. Burgard. Where is ...? Learning and utilizing motion patterns of persons with mobile robots. In *Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 909–914, Acapulco, Mexico, 2003.
- [5] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. *A System for Video Surveillance and Monitoring*. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.
- [6] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.
- [7] A. Galata, N. Johnson, and D. Hogg. Learning variable length Markov models of behaviour. *Computer Vision and Image Understanding Journal*, 81:398–413, 2001.
- [8] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.
- [9] 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.
- [10] L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proceedings of the National Conference on Artificial Intelligence(AAAI-04)*, 2004.
- [11] N. Nguyen, H. Bui, S. Venkatesh, and G. West. Recognising and monitoring high-level behaviours in complex spatial environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
- [12] N. Nguyen, S. Venkatesh, G. West, and H. Bui. Learning people movement model from multiple cameras for behaviour recognition. In *Joint IAPR International Workshops on Structural and Syntactical Pattern Recognition and Statistical Techniques in Pattern Recognition (SSPR2004)*, pages 315–324, Lisbon, Portugal, 2004.
- [13] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Fourth IEEE International Conference on Multimodal Interfaces*, pages 3–8, 2002.
- [14] S. Osentoski, V. Manfredi, and S. Mahadevan. Learning hierarchical models of activity. In *IEEE/RSJ International Conference on Robots and Systems (IROS)*, 2004.
- [15] D. Pynadath and M. Wellman. Generalized queries on probabilistic context-free grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):65–77, 1998.
- [16] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
- [17] G. Wei and M. Tanner. A Monte-Carlo implementation of the EM algorithm and the poor man’s data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990.