

# Recognition of Human Activity through Hierarchical Stochastic Learning

Sebastian Lühr    Hung H. Bui    Svetha Venkatesh    Geoff A. W. West  
School of Computing  
Curtin University of Technology  
GPO Box U1987 Perth, 6845 Western Australia  
E-mail: {luhrs, buihh, svetha, geoff}@cs.curtin.edu.au

## Abstract

*Seeking to extend the functional capability of the elderly, we explore the use of probabilistic methods to learn and recognise human activity in order to provide monitoring support. We propose a novel approach to learning the hierarchical structure of sequences of human actions through the application of the hierarchical hidden Markov model (HHMM). Experimental results are presented for learning and recognising sequences of typical activities in a home.*

## 1. Introduction

It is no secret that as we age our cognitive and physical abilities slowly deteriorate and we grow dependant on others for care, normally through nursing homes, visiting day carers or our relatives. We wish to offer the elderly an alternative to such third party care and enable them to maintain their independence so that they may age gracefully within their own homes. We seek to achieve this by creating intelligent environments that are capable of learning their occupants' normal behavioural patterns so as to recognise abnormalities and to take appropriate action; using visual and audio cues to help them or summon outside assistance in a medical emergency or a potentially dangerous situation.

A prerequisite to this task is the learning of models of different activities carried out by people in their daily schedule. Once such models are learned, they can be used in classifying new activities or detecting abnormalities.

In this paper, we have applied the hierarchical hidden Markov model (HHMM) [3] to the task of learning the probabilistic nature of simple sequences of activities such as in the preparation of an evening meal. The HHMM was chosen because, as with the hidden Markov model (HMM), its statistical nature offers the ability to deal with noisy and missing data. Unlike normal HMMs, however, the HHMM allows us to generate models whose structure can capture the natural hierarchy present in human activity, thus mak-

ing it easier to interpret and answer queries at varying levels of abstraction. Furthermore, the hierarchical nature of the HHMM allows for model reusability and faster learning through independent sub-model generation [10].

Although the HHMM has been used to learn models for handwritten letter recognition and robot navigation, we are among the first to propose the use of the HHMM in modelling human activity. The hierarchical nature of the HHMM facilitates the creation of complex models that incorporate patterns over the short, medium and long term at varying levels of resolution and abstraction. Rather than recognising gestures, activities or atomic events, we seek to learn overall behavioural patterns of a person.

The organisation of this paper is as follows: In the next section we discuss work related to the recognition of human activity. We describe our approach using the HHMM in Section 3. The experiments and their results are presented in Section 4. Finally, we conclude with a summary of the work undertaken and highlight the issues and limitations that are still required to be overcome.

## 2. Related Work

The HHMM was first proposed in [3] and is a special case of stochastic context free grammar (SCFG). The authors applied the model to handwriting detection, demonstrating that a HHMM trained on a single handwritten word is able to learn the hierarchical nature of the training data, populating the lowest states in the model topology with observations mapping to atomic strokes, compound strokes representing letters and combinations of letters at the middle layers and finally the word itself at the top level. The concept of hierarchical HMMs has since been applied in the extension of partially observable Markov decision processes (POMDPs) to hierarchical POMDPs (HPOMDPs) and used in indoor environment learning for robot navigation [10].

Similar to the HHMM, the abstract hidden Markov model (AHMM) [2, 1] is a hierarchical stochastic model

for representing a hierarchical abstraction of an agent's state and goals at varying levels of detail. The AHMM extends the HHMM by allowing the refinement of a layer into sequences at the lower level to be dependent on the current "environment" state. This allows for simple types of context-sensitive behaviours to be modelled. Inference scalability in the AHMM is attained by limiting interaction between chains to only those directly above or below it. The AHMM has recently been applied to tracking behaviour and recognising activities being carried out by two people in a computer vision laboratory [6]. In this experiment the authors were able to successfully distinguish between a person interacting with an object and only passing by the object, predicting the subjects' intentions only through observations of their location.

Using HMMs and coupled HMMs (CHMMS) [7] built models of peoples' individual and interactive behaviour based on their proximity and trajectories in an outdoor scene. The models were initially trained using data produced by synthetic agents and then refined using data collected from the real-world scene under surveillance.

Stochastic context free grammars were employed by [4] to recognise activities. They used HMMs to detect the occurrence of low-level events representing the primitives used in their grammar parser.

### 3. The hierarchical hidden Markov model

In this section we briefly introduce the hierarchical hidden Markov model. We follow the notation used in [3] which defines an observation sequence,  $\overline{O} = [o_1, o_2, \dots, o_T]$ , to be a finite length string from all possible strings,  $\sum^*$ , from the finite alphabet  $\sum$ . States within the HHMM are represented by  $q_i^d$  where  $d \in [1, 2, \dots, D-1, D]$  denotes the hierarchy level and  $i$  the state index relative to the parent. The state index may be omitted if it is clear which state is being referred to. States are one of three types; internal, end or production. Internal states are themselves HHMMs and may have an arbitrary number of children states, the number of non-end sub-states of state  $q_i^d$  being denoted by  $|q_i^d|$ . Production and end states do not have children. Vertical and horizontal transition probabilities are defined for each internal state as the vector  $\Pi^{q^d}$  and the matrix  $A^{q^d}$  respectively. An internal state must always perform a vertical transition down to one of its children before a horizontal transition may be made, control of the transitions returning to the calling state only when a lower state has made a horizontal transition to an end state. The end state is a special token state that exists only to signal when an upwards vertical transition is to be made. The probability of state  $q^{d-1}$  vertically transitioning to sub-state  $q_i^d$  is specified as  $\pi^{q^{d-1}}(q_i^d)$  while the probability of state  $q_i^d$

making a horizontal transition to state  $q_j^d$  is written as  $a_{ij}^{q^d}$ . Production nodes are the only states within the HHMM that emit observations and are much like the states of a HMM. The discrete probability density function (PDF) of the production nodes is represented as the vector  $B^{q^D}$  which defines the probability of state  $q^D$  producing observation  $v_k$  as  $b^{q^D}(k)$ . The model parameters are denoted in the compact form  $\lambda = (A, B, \Pi)$ .

To compliment the forward,  $\alpha$ , and backward,  $\beta$ , path variables of the HMM, the HHMM introduces two new variables,  $\chi$  and  $\xi$ , corresponding to the downward and upward transition probabilities respectively. The notation and meaning of the path variables is given by

$$\alpha(t, t+k, q_i^d, q^{d-1}) = P(o_t \dots o_{t+k}, q_i^d \text{ completed at } t+k | q^{d-1} \text{ started at } t) \quad (1)$$

$$\beta(t, t+k, q_i^d, q^{d-1}) = P(o_t \dots o_{t+k}, |q_i^d \text{ started at } t, q^{d-1} \text{ completed at } t+k) \quad (2)$$

$$\xi(t, q_i^d, q_j^d, q^{d-1}) = P(o_1 \dots o_t, q_i^d \text{ transitions to } q_j^d, o_{t+1} \dots o_T | \lambda) \quad (3)$$

$$\chi(t, q_i^d, q^{d-1}) = P(o_1 \dots o_{t-1}, q^{d-1} \text{ transitions to } q_i^d, o_t \dots o_T | \lambda) \quad (4)$$

For complete definitions and derivations of the path variables and their auxiliary variables see [3].

To allow for multiple observation sequences, we have modified the original re-estimation formula based on [8]. The set of  $N$  observation sequences is defined as  $O = [O^1, O^2, \dots, O^N]$  and we refer to a single observation in  $O^n$  as  $o_t^n$ . As the sequence strings are independent, the likelihood of all observation sequences being produced by the model is

$$P(O|\lambda) = \prod_{n=1}^N P(O^n|\lambda). \quad (5)$$

We shall use the short hand notation

$$P_n = P(O^n|\lambda). \quad (6)$$

We can define the new vertical and horizontal transition probability estimation as a normalised sum of the individually weighted observation sequences as follows:

$$\hat{\pi}^{q^1}(q_i^2) = \frac{\sum_{n=1}^N \frac{1}{P_n} \chi(O^n, 1, q_2^2, q^1)}{\sum_{n=1}^N \frac{1}{P_n} \sum_{i=1}^{|q_i^1|} \chi(O^n, 1, q_2^2, q^1)} \quad (7)$$

$$\hat{\pi}^{q^{d-1}}(q_i^d) = \frac{\sum_{n=1}^N \frac{1}{P_n} \sum_{t=1}^T \chi(O^n, t, q_i^d, q^{d-1})}{\sum_{n=1}^N \frac{1}{P_n} \sum_{i=1}^{|q_i^{d-1}|} \sum_{t=1}^T \chi(O^n, t, q_i^d, q^{d-1})} \quad (8)$$

$$\hat{a}_{ij}^{q^{d-1}} = \frac{\sum_{n=1}^N \frac{1}{P_n} \sum_{t=1}^T \xi(O^n, t, q_i^d, q^{d-1})}{\sum_{n=1}^N \frac{1}{P_n} \sum_{t=1}^T \gamma_{out}(O^n, t, q_i^d, q^{d-1})}. \quad (9)$$

The re-estimation of the observation matrix is presented as equation 10.

## 4. Experimental Results

We implemented a robust motion tracker using a Gaussian background model for foreground segmentation as described in [9] and a Kalman filter to track objects across frames [5]. The tracker returns the world coordinates of the feet of the person under surveillance which we map onto discrete observations based on the person’s Euclidean distance to predefined areas of interest; the door, fridge, food preparation area, sink and the stove in our kitchen environment and the door, dinning table, television, bookcase and couch in our lounge room environment. An undefined label is assigned when the person is not in close proximity to any of the defined areas.

Four simplistic styles of dinner preparation and five typical lounge room activities were recorded several times at 25fps, each over a period of 60 to 70 seconds. The tracker extracted an observation sequence for each of the recordings by sampling the person’s location every 25 frames.

The first cooking style, Figure 1(a), involves spending some time preparing the food and rummaging through the fridge before the meal is cooked. The second cooking style, Figure 1(b), involves washing dishes prior to cooking on the stove. The third cooking style has the person first going to the sink to wash the dishes then spending time at the food preparation area and the fridge before finally cooking the meal. The last cooking style sees the subject transitioning between each area of interest in a round robin fashion, starting and ending at the stove, before leaving the room.

Similarly, the lounge room sequences featured transitions between areas of interest representative of typical lounge room activities, which we called “watch television”, “read a book on the couch”, “eat dinner”, “eat dinner while watching television” and “there is nothing good on TV, read a book instead.”

A total of four cooking and five lounge room models were built using the method described in Section 4.1 from a dataset of 36 training sequences, four per model, consisting of 16 cooking sequences and 20 lounge room sequences. In Section 4.1 and Section 4.2 we only discuss the HHMM’s ability to learn and represent the first two cooking sequences. Classification results from all models, however, are presented in Section 4.3.

### 4.1. Learning Individual Styles

In our first experiment we built a separate model for each of the first two cooking styles using a three layer topology. The topology used, depicted in Figure 2 and Figure 3, features all the production states at the lowest layer. Three internal states on the second level govern the vertical transitions to, and horizontal transitions among, the production nodes. Finally, the root node on the top level regulates the activation of, and horizontal transitions among, the second level internal states.

Examining the first model, we notice the PDFs of the production states closely associate their calling parent states with clearly definable activity regions on our sequence maps. For example, the production nodes on the leftmost sub-tree only emit the `Near Stove` and `Undefined` labels. Given the context, we can associate the `Undefined` label with a person going to and from the stove area, as a person acting out the first style needs to first traverse the middle of the room and in doing so will not be in close proximity to the predefined areas. We hence label the internal parent state in Figure 2 as “Cooking.”

The production state children of the second internal state primarily emit `Near Fridge` and `Near Food Prep` observations. The transition structure among the production nodes suggests a strong dependence between these two observations which closely corresponds to the “Fridge then food preparation then fridge then food preparation” loop present in the training sequences. We have labelled this state “Fridge & Food Prep.” Analysis of the third internal state’s children reveals that the internal node is responsible for generating only `Near Door` observations. As this state is the only one vertically transitioned to by the root node and also the last internal node to be activated prior to termination of the entire model, we assign this internal state the label “Enter/Exit.”

Tracing and labelling the state transitions governed by the root node we find that the entire dinner preparation sequence is abstracted as “Enter/Exit → Fridge & Food Prep → Cooking → Enter/Exit” and corresponds almost completely with the atomic observations of the training data.

Similar results emerged through visual analysis of the second model in Figure 3. The three internal states on the

$$\hat{b}_{q_i^D}^{q^{D-1}}(v_k) = \frac{\sum_{n=1}^N \frac{1}{P_n} [\sum_{o_i^n=v_k} \chi(O^n, q_i^D, q^{D-1}) + \sum_{t>1, o_i^n=v_k} \gamma_{in}(O^n, t, q_i^D, q^{D-1})]}{\sum_{n=1}^N \frac{1}{P_n} [\sum_{t=1}^T \chi(O^n, q_i^D, q^{D-1}) + \sum_{t=2}^T \gamma_{in}(O^n, t, q_i^D, q^{D-1})]} \quad (10)$$

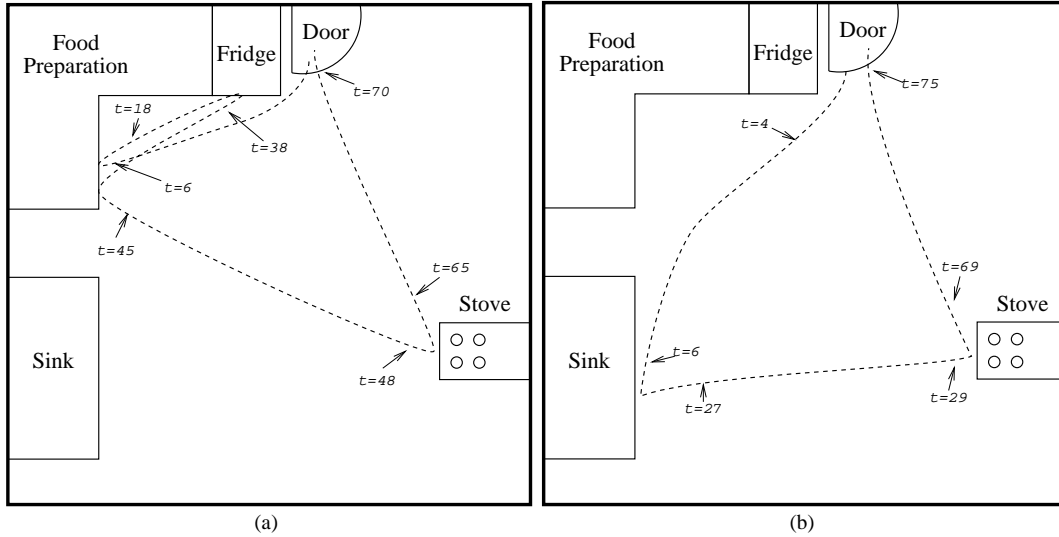


Figure 1. Layout of our kitchen showing the approximate track and elapsed time,  $t$ , in seconds for (a) the “food preparation first” and (b) the “washing dishes first” meal preparation sequences.

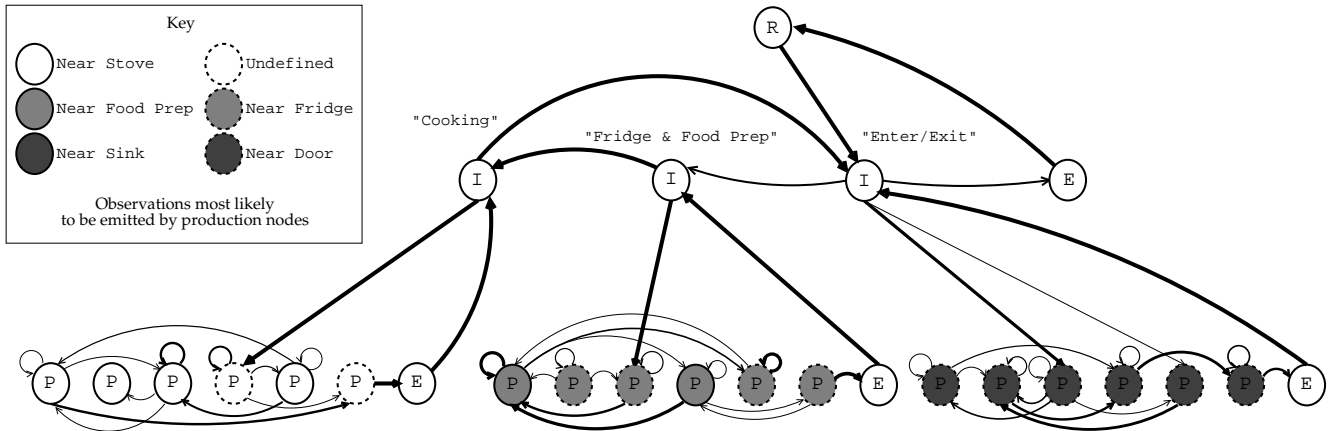


Figure 2. The learned model for the “food preparation first” style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

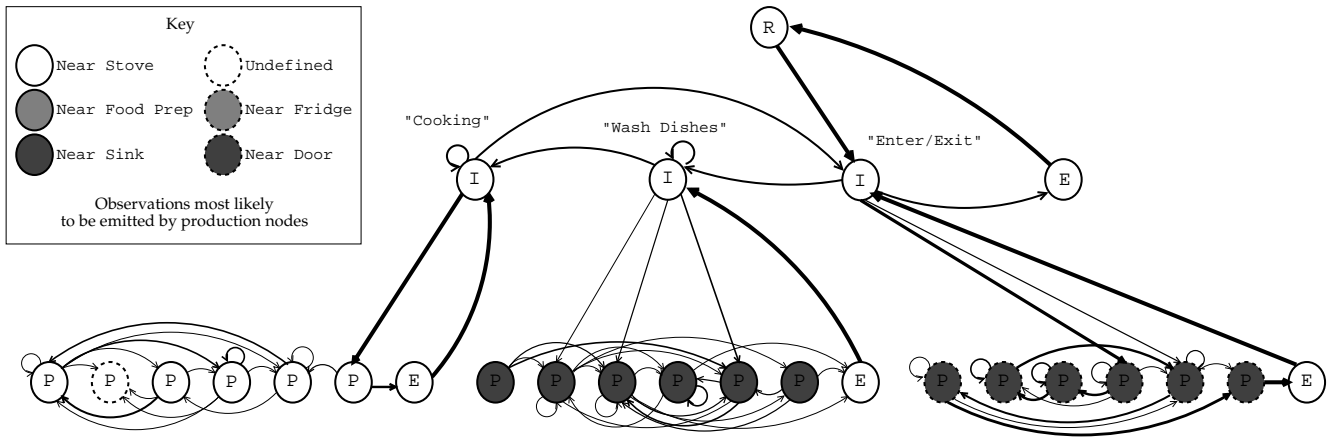


Figure 3. The learned model for the “washing dishes first” style. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

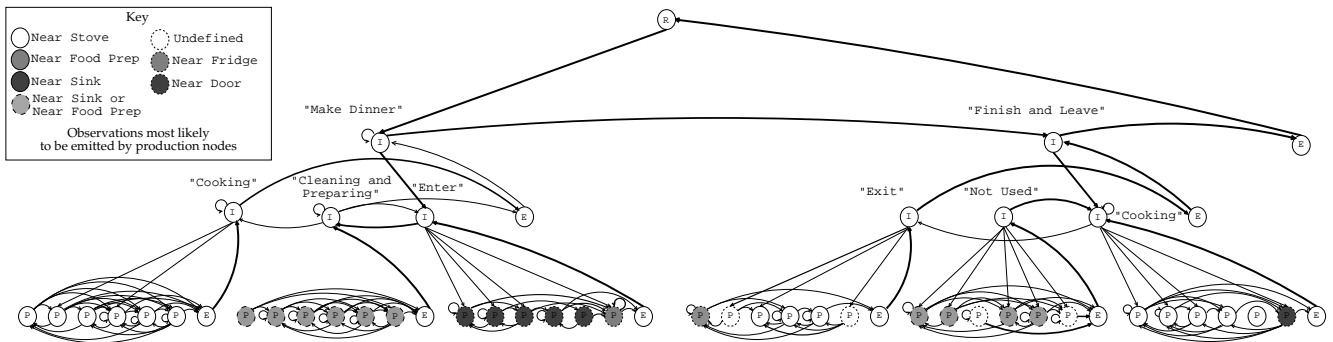


Figure 4. The learned model for the combined styles. Darker arrows indicate stronger state dependence and a higher transitional likelihood among the production (P), internal (I) and end (E) states. Insignificant transitions have been omitted for clarity.

second level were assigned the labels “Cooking,” “Wash Dishes” and “Enter/Exit” for the observations emitted by their respective production sub-states. Again, the transitional structure of the internal states strongly portrays the activity in the training data. Comparing the two models, it is interesting to note the presence of self-transitions in the second model that are absent from the first. Whereas the production nodes in the first model have a clear and structured transition structure this structure is not as clearly defined in the second model. This suggests that the production nodes in the first model possess the entire state duration information while the internal nodes of the second model have heavily adopted this role.

## 4.2. Learning Both Styles

In our second experiment we trained the four layer topology in Figure 4 on the entire observation sequence set. The new topology now features six internal states on the third level, their activation controlled by two internal nodes on the level above. By examining the observations likely to be emitted by the production states, we were able to assign the first state on level three the label “Cooking.” The next internal state was given the label “Cleaning and Preparing” given the evenly distributed likelihood of producing the Near Sink and Near Food Prep observations.

The third, fourth and sixth product state parents were defined as “Enter,” “Exit” and “Cooking.” In defining their meaning we took the transitional structure of the first and second layers into account to distinguish the Enter and Exit states rather than the combined Enter/Exit nodes previously encountered in the first experiment. By noticing that the model will first activate the Enter state and is highly likely to terminate only after the Exit sub-states are executed, we assume that the model differentiates between the two events. The fifth state on the same level was designated “Not Used” as no transitions to this state are ever made.

Using the meaning now associated to the above states we could then assign meaningful labels to their parents on the second level. As the first of these states shows a strong likelihood of the sub-states being activated in the order Enter then Cleaning and Preparing followed by a reasonable likelihood of Cooking, we assign it the label “Make Dinner.” The probability of the Cleaning and Preparing transitioning to the neighbouring Cooking label appears unusually low. This is acceptable, however, considering that a Cooking state is also present in the second half of the topology and is the first to be activated following termination of Making Dinner. The last internal state is labelled “Finish and Leave” as it accounts for the conclusion of the training data, the

cooking of the meal and exiting of the room.

Tracing the state activations we can observe the model abstracts the two cooking styles as the sequence Making Dinner → Finish and Leave. This can be further expanded as Enter → Cleaning and Preparation → Cooking → Exit which provides a good abstraction of the combined dinner preparation styles.

One observation that can be made is that too many degrees of freedom are afforded by the topology resulting in a model that combined the two styles, spreading their structure over the entire topology rather than learning and separating the two styles. This problem could be overcome through model re-use, helping to create richer models by fixing the parameters of previously learned models as discussed in [10].

## 4.3. Classification Results

Twenty seven unseen sequences were used to test the ability to classify activity given the models. The cooking and lounge room models were only tested on cooking and lounge room sequences respectively as the person’s locality indicates, in this particular case, the models of interest. The test sequences were designed to include major variations of duration and, in some cases, variations in transition order between the areas of interest.

The classification results of the cooking and the lounge room activity test sequences, presented in Table 1 and Table 2 respectively, demonstrate reasonable classification accuracy. These preliminary results demonstrate that generally the classification is correct. Current work is focused on further validation and investigation of the misclassification.

**Table 1. Confusion matrix for the cooking test sequences.**

	Prepare, cook	Dishes, cook	Wash, prepare, cook	Round robin
Prepare, cook	1	0	1	1
Dishes, cook	0	3	0	0
Wash, prepare, cook	0	0	3	0
Round robin	0	0	0	3

## 5. Conclusion

In this paper, we have briefly introduced the HHMM and documented the modification necessary to support multiple observation sequences in training. Using the HHMM we have shown that, given a three layer topology, we are able

**Table 2. Confusion matrix for the lounge room test sequences.**

	Watch TV	Read (couch)	Dinner	TV dinner	TV then read
Watch TV	3	0	0	0	0
Read on couch	0	3	0	0	0
Dinner	0	0	1	2	0
TV dinner	0	0	1	2	0
TV then read	0	0	0	0	3

to learn abstractions of two simple activity sequences and capture the hierarchical structure present therein.

Such systems will form the foundation for intelligent spaces that can extend the functional capacity of the elderly.

## References

- [1] H. H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden Markov model. *Journal of Artificial Intelligence Research*, To appear.
- [2] H. H. Bui, S. Venkatesh, and G. 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.
- [3] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32:41–62, 1998.
- [4] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, August 2000.
- [5] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [6] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Hierarchical monitoring of people’s behaviours in complex environments using multiple cameras. *International Conference on Pattern Recognition*, August 2002.
- [7] N. M. Oliver, B. Rosario, and A. P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.
- [8] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
- [9] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.
- [10] G. Theodorou. *Hierarchical Learning and Planning in Partially Observable Markov Decision Processes*. PhD thesis, Department of Computer Science and Engineering, 2002.