

Recognising Agent Behaviour During Variable Length Activities

Rolf Baxter and David Lane and Yvan Petillot¹

Abstract. In this paper we present a new method for obtaining situation awareness via the automatic recognition of agent behaviours. In contrast to many other approaches, the presented method models different behaviour durations without using a fixed classification window, and does not require a distribution of behaviour durations. We introduce the Variable Window Layered Hidden Markov Model (VW-LHMM) as an extension of the LHMM to specifically address behaviours with irregular duration. We validate our approach by simulating three high-level behaviours within the harbour and coastline security domain. We compare performance against the LHMM and show that our approach provides a 10% improvement in classification accuracy, in addition to earlier classification.

1 INTRODUCTION

In the academic literature several terms are used to refer to behaviour recognition. These include activity recognition, plan recognition and goal recognition. This paper will draw the distinction that activity recognition involves short term behaviours with an immediate goal. For example, *walk in a straight line from A to B*. In contrast to this, plan and goal recognition identify long term goals such as the military objective *infiltrate compound C*. This paper will refer to both these levels of abstraction individually, and use behaviour to refer to the overall process.

To classify an agent's behaviour from observations, probabilistic models compare the competing hypotheses to determine the most likely class. The observation sequence is then truncated and the process repeats. However, physical behaviours are often similar for portions of their duration. For example, the behaviours *MoveForward* and *PerformUTurn* both involve forward motion so cannot be immediately distinguished. These behaviours can therefore only be identified once their activity diverges.

Although this in itself is not a challenge, many behaviours also have irregular durations. For example, assume that *MoveForward* can take [0:10] steps, while *PerformUTurn* involves [0:5] steps forwards, followed by [0:2] steps right and [1:5] steps back. Under these conditions, an observation sequence cannot be classified until at least six time-steps have passed, because identical observations can be generated by either behaviour for the first five time-steps. However, there is also the potential that a behaviour will be identifiable after only one time-step, such as when a step backwards is observed at time-step one. In this example it is clear that a fixed size window prolongs classification. For most applications of on-line behaviour recognition, early classification is a desirable trait, however, the aforementioned approach does not facilitate this goal.

Some researchers have proposed approaches that dynamically identify when a behaviour terminates. Bui and Duong used termination distributions [4, 2], although recognition was based on agent trajectories within known environments and cannot be directly applied to less constrained problems. Coastline security may involve unknown environments, preventing fixed agent trajectories from being learnt. Furthermore, it is unclear how these approaches perform when the termination distributions are extremely broad.

In this paper we extend the Layered Hidden Markov Model (LHMM) to directly address the early recognition of variable duration behaviours. Furthermore, because our recognition is based upon spatially unconstrained trajectories, our model does not need to be retrained when moving from one environment to another.

Instead of classifying observations every T time steps, or when some termination state is reached, our approach classifies behaviour whenever a clear difference in likelihood is apparent. Behaviours that continue to exhibit similarities are observed for longer periods of time, while vastly different behaviours may be classified almost immediately. Experimental results recognising simulated underwater vehicle manoeuvres show that our approach is able to recognise behaviours both more accurately, and earlier than the original LHMM.

The next section will discuss related work in the area of behaviour recognition. As the foundation for our model, the LHMM will be introduced in Section 2. This will be followed by a discussion of the model's limitations in Section 2.1, and an introduction to our extension: the Variable Window Layered Hidden Markov Model (VWLHMM). Section 3 will introduce the validation domain and will be followed by a discussion of the results, and Section 4 will conclude and discuss our plans for future work.

1.1 RELATED WORK

Contributions to behaviour recognition research have come from a broad range of communities. Computer vision researchers such as Oliver have focused upon understanding simple human activities from video [6, 11], while artificial intelligence researchers such as Kautz have often focused on recognising high level, abstract behaviours [1, 5, 7, 13]. Some previous work can also be categorised as cross-community, such as the recognition of abstract, high-level behaviour from low-level video [2, 9, 10].

Although video has proven a popular source of agent observations, other researchers have used GPS locations to track agent movements. Liao recognised agent behaviours within a city by concatenating an agent's location with supplementary data such as the location of bus stops [8]. Similarly, Chai and Yang predicted an agent's goal by using WIFI signal strength to track an agent within a known indoor environment [3].

¹ Ocean Systems Laboratory, Heriot-Watt University, UK, email: {R.Baxter, D.M.Lane, Y.R.Petillot}@hw.ac.uk

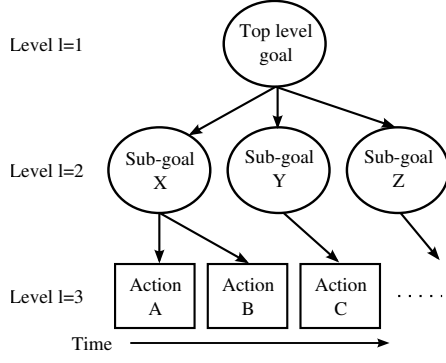


Figure 1. A Hierarchical Task Network

To represent a behaviour, its environment and temporal characteristics can be dynamically modelled using techniques such as the Hidden Markov Model [12]. However, it has been recognised that complex behaviours often involve long-term temporal constraints that are inconsistent with a Markov approach [4]. This has led to the development of hierarchical models that reason at multiple levels of abstraction. For instance, Oliver *et al.* used a different temporal window at each level of their Layered Hidden Markov Model. This allowed low-level activities to be detected over shorter durations, while high-level activity evolved more slowly.

Other researchers have modelled behaviour duration dynamically by identifying when behaviours terminate. In the Hierarchical Hidden Markov Model behaviours terminate when an agent reaches a designated landmark [10]. In the Abstract Hidden Markov Model behaviour termination is modelled using a probability distribution over a set of states [2]. Additionally, both of these models adopt hierarchical structures and therefore model both long term temporal dependencies and variable behaviour duration. Unlike this previous work, our validation does not use geographical states and thus once a model has been trained, can recognise agent manoeuvres in any geographical location.

2 LAYERED HIDDEN MARKOV MODELS

To classify the underlying intent of an observation sequence it is helpful to first invert the problem. Consider the tree structure shown in figure 1 in which a three-level Hierarchical Task Network (HTN) is shown. Here, a top-level goal is being decomposed into three sub-goals (activities), and each sub-goal is decomposed into a finite set of actions that achieve that sub-goal.

A top level goal can be modelled by a set of states representing sub-goals and actions. A state is denoted as p and each goal state is termed an abstract state, while actions are termed production states. For each state, a set of child states, denoted $ch(p)$, determine the actions and goals that support the completion of the parent state p .

For each state and its set of children $ch(p)$, denote π^p as the prior distribution over $ch(p)$, and $\alpha_{i,j}^p$ as the transition probability from child i to child j : $i, j \in ch(p)$. Finally, $E_e(p)$ denotes the probability of observing state e given that state p is being performed. The entire HTN can thus be represented by the set of parameters $\theta : \{ \pi^p, \alpha_{i,j}^p, E_e(p) : \forall i, j, e, p \}$.

An HTN of depth D can be modelled by a hierarchy of $D-1$ Hidden Markov Models (HMMs) and the θ parameters. For clarity, the bottom of the hierarchy will be described in isolation first.

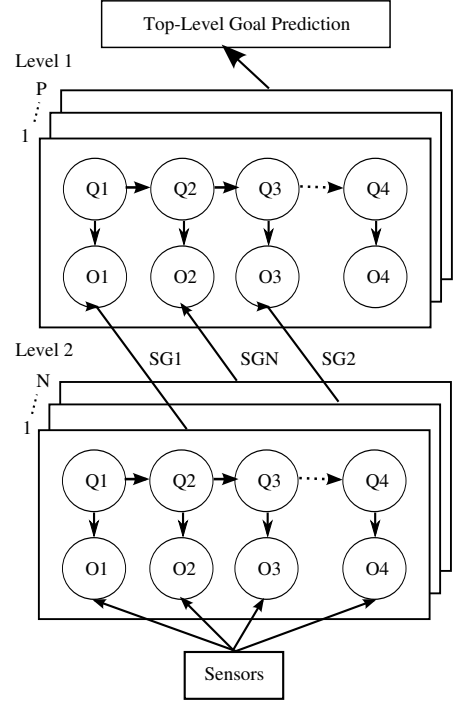


Figure 2. The hierarchical structure of the Layered Hidden Markov Model (LHMM)

A sequence of observations $O = (o_1, o_2, \dots, o_n)$ is observed such that each $o_i \in \gamma$, where γ is the set of production states. For each observation o_t we assume that the true state (q_t) is affected by noise. This is modelled as $p(o_t = \gamma_e | q_t = \gamma_i)$. Each abstract state k can thus be modelled by an HMM λ_k with parameters $\theta^* : \{ \pi_k^p, \alpha_{k,i,j}^p, E_{k,e}(i) : \forall i, j, e, p \}$, in which $E_{k,e}$ are the emission probabilities $E_{k,e}(i) = p(o_t = \gamma_e | q_t = \gamma_i)$.

For each remaining level in the HMM hierarchy, the abstract states are modelled in the same way, with the exception that the observations at higher levels are abstract states emitted by the level below, rather than directly observed production states.

The HMM hierarchy can thus be referred to by a vector ζ of size $D-1$, where each element contains a set of HMMs λ^l representing a certain level in the hierarchy.

For the LHMM in [11], state durations are fixed for each level in ζ . This means that every W^l observations, the state representing the most likely model in λ^l is appended to the end of observation sequence O^{l-1} . That is with the exception of when $l = 1$, where the output state becomes the top-level classification. Recognition is therefore performed in a bottom-up approach with lower level classifications feeding up the HMM hierarchy. This process is illustrated in figure 2, in which the two-level hierarchy classifies observations by sub-goal and top-level goal. This hierarchy is in fact the reverse of the HTN in figure 1, where level three in the HTN is directly observed by sensors.

To map figure 2 back to figure 1 using a concrete example, assume that we observe actions a , b and c , and that the fixed temporal window at the bottom of the hierarchy is two time-steps. After observing a and b , each level two HMM calculates the probability of the observed sequence. The most probable HMM is then selected, which in

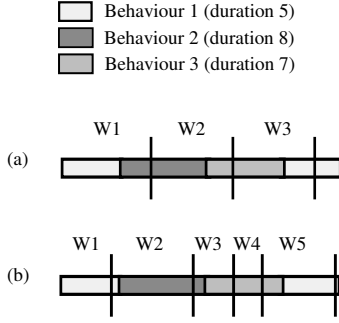


Figure 3. (a) Equal size temporal windows W1-W3 fall out of sync with behaviour 1-3. (b) Windows increase and decrease according to confidence in behaviour detected

this case should be the HMM representing *Sub-goal X*. A new observation representing this sub-goal is then generated and forms a new observation for the level one HMMs. This process is then repeated, where each level one HMM represents a different top-level goal. The most probable level one HMM thus becomes the top-level goal classification of the observed behaviour.

2.1 Activities and Goals with Variable Durations

A drawback of the LHMM is illustrated in figure 3a, in which there are 3 behaviours along a time line. The duration of each behaviour is represented by the length of a rectangle, while W1-W3 represent the temporal windows determined by W^l . By having a fixed window size, classification is forced to occur during behaviour transitions. For example, in W3, behaviours 1 and 3 are observed for equal durations so a correct classification is difficult. As a result, it is likely that segments of the sequence will be incorrectly classified.

If behaviour durations were consistent within a particular observation sequence, the temporal window in [11] could be adapted to match the observed durations. However, this cannot be performed when durations are inconsistent, such as when a behaviour is observed for six time steps on its first occurrence, and 16 on its second.

To overcome this problem, the Variable Window Layered Hidden Markov Model (VWLHMM) replaces the temporal window W^l by the tuple $\langle R^l, C^l, M^l \rangle$. These parameters represent the earliest point a classification can be made (C^l), the latest point (M^l), and a certainty parameter (R^l).

To illustrate how the parameters are used to adapt the classification window, consider the classification process for an observation sequence O^l . Model $\lambda_k^l \in \lambda^l$ is selected as the most probable source of $O_{t:T}^l = f_{t,T}(O^l) = (O_t^l, \dots, O_T^l)$ according to one of two variants below. In both variants, $\Theta(O|\lambda_k^l)$ is the probability $p(O|\lambda_k^l)$ normalised to be in the range [0,1] by dividing by $\sum_i p(O|\lambda_i^l) \forall \lambda_i^l \in \lambda^l$.

1. VWLHMM (R) Ratio variant

Model $\lambda_k^l \in \lambda^l$ is the most probable source of $O_{t:T}^l$ iff :

$$\Theta(O|\lambda_k^l) > R^l * \max(\Theta(O|\lambda_i^l)) \forall \lambda_i^l \neq \lambda_k^l \in \lambda^l \quad (1)$$

Where R^l is some ratio parameter ≥ 1 .

2. VWLHMM (T) Threshold variant

Model $\lambda_k^l \in \lambda^l$ is the most probable source of $O_{t:T}^l$ iff :

$$\Theta(O|\lambda_k^l) > \max(\Theta(O|\lambda_i^l)) \forall \lambda_i^l \neq \lambda_k^l \in \lambda^l \wedge \Theta(O|\lambda_k^l) > R^l \quad (2)$$

Where R^l is some likelihood threshold in the range [0,1].

In both variants, the sequence $O_{t:T}^l$ is classified as model λ_k^l when the probability constraint is satisfied for C^l consecutive observations. Or in other words, whenever that model becomes some magnitude more probable than the rest, and remains at least that probable for C^l observations.

If the constraint is satisfied after x observations, the observation sequence $(O_t^l, \dots, O_{t+x}^l, O_{t+x+1}^l, \dots, O_N^l)$ is truncated to $(O_{t+x+1}^l, \dots, O_N^l)$ and a classification state is emitted to level $l-1$. If the classification constraint has not been satisfied after M^l observations, the observation sequence $(O_t^l, \dots, O_{t+M^l}^l, O_{t+M^l+1}^l, \dots, O_N^l)$ is truncated to $(O_{t+M^l+1}^l, \dots, O_N^l)$, and no state is emitted.

Figure 3b illustrates the result of VWLHMM classification, in which temporal windows now change size as observations are made. For example, in W2 further observations are required to correctly identify behaviour 2, leading to a longer temporal window than W1. The LHMM would be forced to make an early classification, however, the VWLHMM is able to increase the temporal window to consider more of the observed behaviour. This increases the probability of correct classification. Similarly, the temporal windows are all smaller than in figure 3a because they satisfy the classification criteria. In other words, early classification is allowed whenever one model is significantly more probable than any other.

An agent's goal is recognised by considering each bank of HMM parameter sets as classifier parameters. To classify a sequence of sensor observations O the algorithm starts at level $l = D - 1$ and proceeds by calculating $p(O^l|\lambda_k^l)$ for each model in λ^l , where $O^{l=D-1} = O$. This is performed using the forward element (α) of the Forward-Backward procedure from [12]. Upon calculating the probability of O^l being generated by each model in λ^l , the classification criteria are checked. If classified, the output classification is then appended to O^{l-1} . At any level, the arrival of a new observation causes the recognition algorithm to step forwards at that level.

2.2 Parameter Learning

Estimating ζ , the HMM parameters for each level in the VWLHMM, requires action sequences in which the underlying goal is both constant and known. Here, we use the term goal to reference the immediate goal at the next level up in the hierarchy. Training therefore commences at the bottom of the hierarchy, using sensor observations with a known immediate goal. Once trained, this level may then be used to generate an abstracted observation sequence by classifying the original sensor input. This provides training data to the next level up in the hierarchy, where the process can then repeat.

One method for estimating these parameters is to use the Baum-Welch algorithm in which the $p(O^l|\lambda_k^l)$ is maximised through iterative update and improvement. We skip the details of this algorithm but refer the interested reader to [12]. Using this algorithm to learn the HMM hierarchy has complexity $\sum_{l=0}^{D-1} O(|N^l|)^2 * |seq^l|$, where $|seq^l|$ is the length of the training sequence at level l and $|N^l|$ is the number of abstract states.

The VWLHMM parameters $\langle R, C, M \rangle$ lie outside the Baum-Welch algorithm so must be arbitrarily initialised, and may be further tuned using optimisation techniques. For the threshold variant, goal likelihood graphs are particularly useful for initialising the R parameter. Example graphs can be seen in figure 4, in which three graphs are shown, one for each top-level goal in the validation corpus. The average point at which the likelihoods diverge is a good estimate for R , and might be set to 0.5 or 0.6 in this example. The parameters

$\langle C, M \rangle$ must also be initialised, and in this work were frequently set to 3 and 10 respectively. In the results reported here, satisfactory values for $\langle R, C, M \rangle$ were discovered by hand within a reasonable time and so were not further optimised.

If the parameters $\langle R^l, C^l, M^l \rangle$ have been incorrectly set there is the potential for all input to be discarded. In this eventuality, increasing M^l may resolve the problem, although in the worst case, input may still be discarded if the behaviours are too similar.

3 Experimental Evaluation

To validate the architecture described above we demonstrate recognition within a situation awareness application. In this context, situation awareness refers to the ability to detect the manoeuvres of other agents through trajectory recognition. One of the advantages of manoeuvre recognition is that only agent locations need to be tracked. This makes manoeuvre recognition ideal for use in difficult real-world environments where high resolution data may be difficult to obtain. Example applications include autonomous ground vehicles performing base protection, and autonomous underwater systems performing harbour security. Our validation evaluates performance in the harbour security domain by distinguishing between three similar underwater vehicle behaviours.

3.1 The Corpus

The corpus consisted of 168 missions, performed using a hydrodynamic vehicle simulator and observed in a sequential manner. Observations consisted of the vehicles x,y,z coordinates, and in addition to the mission corpus, further data was available in the form of partial missions and was used for training the lowest level in the architecture. The missions in the corpus are representative of real underwater vehicle missions and thus encapsulate realistic vehicle trajectories.

For each mission, the vehicle performs behaviours to achieve one of the three top-level goals shown in table 1. For each goal there is a large degree of overlap, for example, the MCM and MCMI behaviours (figures 5a and c respectively) are identical for much of the time, with MCMI containing only one additional sub-goal (Inspection). This sub-goal may not occur until very late in the mission, resulting in identical behaviour for the majority of the observations. The affect of this attribute will be discussed further in section 3.3.

It is also important to note that the simulator ensured that a high degree of variability was observed between the missions using randomly chosen variables from predefined ranges. For instance, the length and number of turns in a PI mission, length, width and number of sweeps in an MCM mission, and the number and location of inspection circles in an MCMI mission.

3.2 Implementation

In table 1 each of the top-level goals is a class label at the first (top) level in the architecture. When $l = 1$, the HMM bank λ^l models transitions between the six level-2 abstract states. In turn, when $l = 2$, λ^l models transitions between the four level-3 production states.

Level 3 itself lies outside the VWLHMM structure and pre-processes observations using relative vehicle motion. Vehicle heading is calculated at 20 second intervals and transitions in heading give rise to *Veer-Left* and *Veer-Right* observations. Inversely, a 40 second period of single heading motion gives rise to a *Straight* observation. Lastly, increases in vehicle depth produce *Dive* observations, where

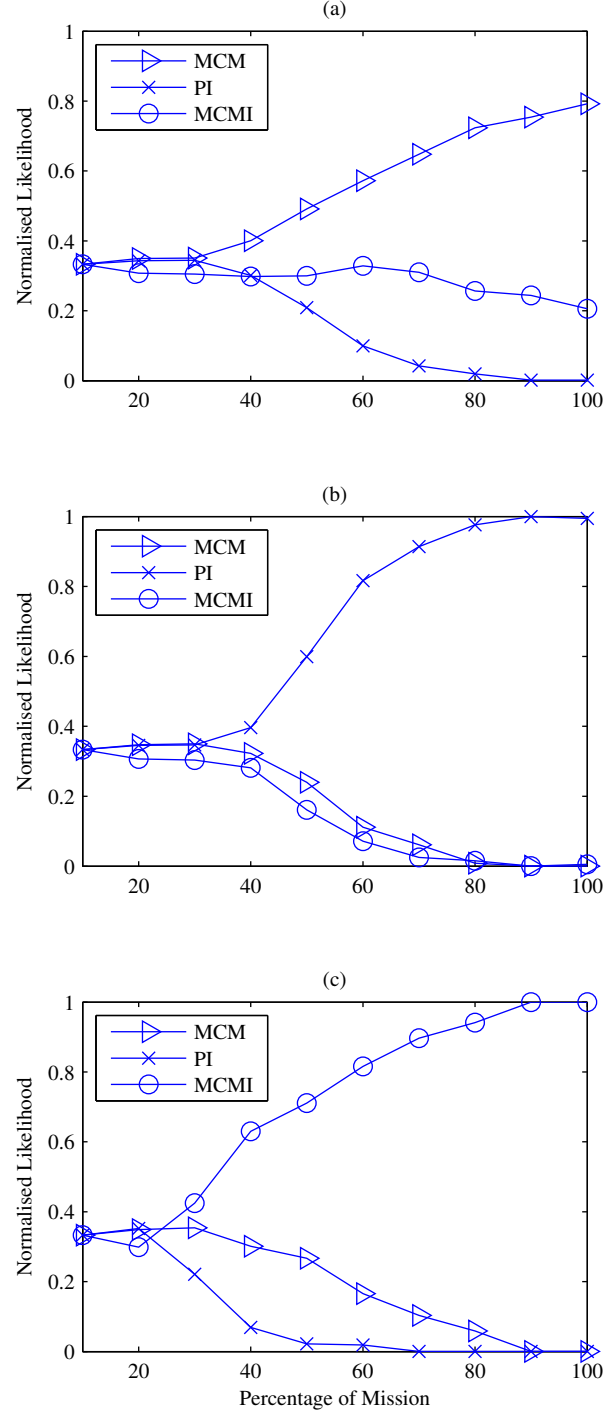


Figure 4. Goal likelihood against % mission progress during an (a) MCM mission, (b) PI mission and (c) MCMI mission

minor fluctuations in vehicle depth are overcome by requiring three increases of depth (each $\geq 30\text{cm}$) across consecutive observations.

In our experiments we compared the performance of three competing models; the original LHMM, the VWLHMM Ratio variant and the Threshold variant. For each experiment the mission corpus was used to perform three-fold cross validation during top-level goal recognition, while level two was trained using separate mission data.

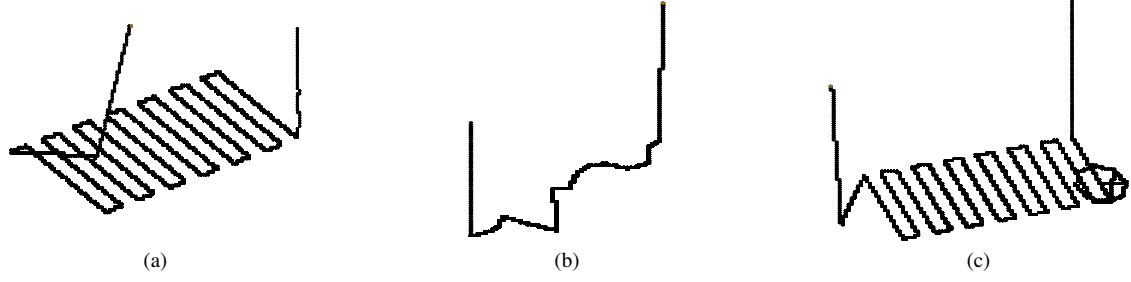


Figure 5. Sample mission trajectories for (a) MCM, (b) PI and (c) MCMI

Table 1. The behaviours recognised at each level in the architecture

Top-Level Goals (Level 1)	Sub-Goals (Level 2)	Actions (Level 3)
MCM	Dive	Straight
MCMI	Track	Veer-Left
PI	Right U-Turn	Veer-Right
	Left U-Turn	Dive
	Inspection	
	End	

The results quantify model performance using the F-score measure; a common statistical technique. The F-Score gives a weighted average of both precision and recall and is calculated as $2(\text{precision} \times \text{recall}) / (\text{precision} + \text{recall})$. Like probabilities, the objective is to maximise the F-Score which is bounded by $[0,1]$. In the context of classification, precision is the number of correct classifications divided by the total number of classifications, while recall is the number of correct classifications over the number of examples.

3.3 Goal Likelihood

The first experiment evaluated the relative likelihood of each goal using the VWLHMM Threshold variant. This was achieved by splitting the corpus into three sections, one for each of the goal behaviours. The architecture was used to calculate the average likelihood of each goal as the missions progressed and are presented in figures 4. Figure 4a illustrates the probability of each behaviour when observing an MCM mission, in which the MCM model has the highest probability, followed by the MCMI model, and lastly the PI model.

Figure 4 shows an interesting phenomenon caused by the similarity of MCM and MCMI missions. During MCMI recognition in figure 4c the probability of the MCM goal decreases more slowly than that of the PI goal. This is because the core behaviour performed during MCM and MCMI is identical, so the correct goal can only be disambiguated upon observing an inspection sub-goal. As the mission time is increased more inspections are observed and the likelihood of the MCM goal reduces.

The inverse of this can be observed in figure 4a where the likelihood of the MCMI mission remains at least 0.2 probable. However, none of the behaviour observed during an MCM mission can eliminate the possibility of an MCMI goal, resulting in non-zero MCMI likelihood. Despite this, there is a clear difference in goal likelihood which facilitates good classification accuracy in the next section.

3.4 Goal Classification

The second set of experiments evaluated the goal classification accuracy of the VWLHMM architecture, and uses the LHMM architecture as a comparative baseline. The simplest evaluation scenario consists of only two potential goals: MCM and PI. The first section of table 2 presents these results and shows that the average classification time of all of the models ranges from 50-53% of the mission. The average model F-Scores are also closely related and range from 0.94 (LHMM) to 0.98 (VWLHMM (T)) and illustrate that there is little gain between the architectures.

However, with a more complex scenario of three goals the individual models exhibit clear differences. In the second section of table 2 the parameters were constrained such that classification was achieved after observing no more than 60% of a mission. This restriction allows the models to demonstrate their ability to meet the difficult requirement of early prediction, a feature that is required by many behaviour recognition applications. It is clear that the Threshold variation of the VWLHMM structure delivers both the earliest classification (0.53) and highest average F-Score (0.85), while the LHMM performs worst by converging at 0.59 and has an average F-Score of 0.77. The VWLHMM is therefore able to provide up to a 10% improvement in classifier accuracy, and a 6% reduction in classification time.

It is worth noting that there is relatively little change in PI model performance between the 2 and 3 goal scenarios, while MCM performance dropped by 13-20%. This is caused by the similarity of these missions and the effect on model likelihood observed in figure 4.

Finally, the last section in table 2 shows the maximum F-scores possible by the models when allowing classification times beyond 60%. None of the models significantly improve in performance, suggesting that near-optimal performance was obtained under the constrained approach. We suggest the reason for this in the next section and propose a potential direction for future work.

The run time efficiency of all three models was approximately equal, owing to the fact that the fundamental architecture is shared by all.

4 CONCLUSION AND FUTURE WORK

This paper described the Variable Window Layered Hidden Markov Model, an extension of the LHMM for recognising behaviours with variable duration. Such behaviours are encountered when observing physically embodied agents, as in our validation domain of automated harbour security. We showed that the fixed temporal window in the LHMM causes misclassification when they become out of sync

Table 2. Comparing the performance of the two VWLHMM variants with the LHMM baseline

	Model	Avg. Classification Time (proportion of mission)	F-Score (MCM)	F-Score (PI)	F-Score (MCM)	F-Score (Average)
2 Goals	LHMM	0.5	0.97	0.93	N/A	0.94
	VWLHMM (R)	0.53	0.97	0.96	N/A	0.97
	VWLHMM (T)	0.52	0.99	0.98	N/A	0.98
3 Goals (Avg. Convergence Time ≤ 0.6)	LHMM	0.59	0.72	0.95	0.65	0.77
	VWLHMM (R)	0.59	0.77	0.97	0.72	0.82
	VWLHMM (T)	0.53	0.82	0.98	0.74	0.85
3 Goals (Avg. Convergence Time > 0.6)	LHMM	0.64	0.75	0.97	0.72	0.81
	VWLHMM (R)	0.61	0.77	0.97	0.77	0.83
	VWLHMM (T)	0.70	0.83	0.96	0.82	0.87

with behaviours, while waiting for behaviours to change results in delayed classification. Instead of fixed temporal windows, our approach allows classifications to be made as soon as a behaviour becomes most likely. Inversely, when behaviours cannot be disambiguated, our approach delays classification, where the LHMM would force a potentially incorrect classification.

We validated our approach in the challenging automated harbour and coastline security domain, where underwater agents were monitored using simulated sonar. The architecture was able to distinguish between three overlapping goals with large variability, using location independent actions. The VWLHMM was able to provide a 10% improvement in accuracy over the original LHMM, and reduced classification time by 6%.

To assess performance further, it is important that future work compares performance of the VWLHMM against Semi-Markov Models, such as the Switching Hidden Semi-Markov Model (SHSMM) [4]. This model combines a hierarchical structure with a semi-Markov approach and is therefore also well suited for recognising irregular activities.

To the authors' knowledge, this is the first application of behaviour recognition to the harbour security domain, and has highlighted several challenges for future work. To improve the classification performance it is essential that similar behaviours are modelled more appropriately. For instance, the target inspection behaviour of an MCM mission should be modelled at the first level of the hierarchy in addition to level two. Top-level recognition would then need to be adapted to identify goal transitions, modelling the MCM goal as an interruption of MCM to perform a target inspection. Indeed, this alteration is also necessary to recognise adaptive agent behaviours, where the active top-level goal changes over time.

In this paper the classification parameters $\langle R^l, C^l, M^l \rangle$ were manually defined and thus another direction for future work is to establish how they might be learnt. For instance, it may be possible to use the earliest point at which behaviours diverge to initialise the minimum classification time C^l . Similarly, the classification ratios/thresholds (R^l) were chosen by evaluating goal likelihood graphs (figure 4) and this may be a suitable starting point for automated learning. Finally, it may be possible to initialise the maximum classification time (M^l) by considering the latest point at which behaviours diverge.

ACKNOWLEDGEMENTS

This work was funded by the UK Ministry of Defence under the Competition of Ideas initiative.

REFERENCES

- [1] Nate Blaylock and James F. Allen, 'Fast hierarchical goal schema recognition', in *Proceedings of the National Conference on Artificial Intelligence*, volume 21, p. 796, (2006).
- [2] Hung H. Bui and Svetha Venkatesh, 'Policy recognition in the abstract hidden markov model', *Journal of Artificial Intelligence Research*, **17**, 451–499, (2002).
- [3] X. Chai and Q. Yang, 'Multiple-goal recognition from low-level signals', in *Proceedings of the National Conference on Artificial Intelligence*, volume 20, p. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, (2005).
- [4] Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh, 'Activity recognition and abnormality detection with the switching hidden semi-markov model', in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pp. 838–845, Washington, DC, USA, (2005). IEEE Computer Society.
- [5] C. Geib and M. Steedman, 'On natural language processing and plan recognition', in *Proceedings of the International Joint Conference on AI*, pp. 1612–1617, (2007).
- [6] Asaad Hakeem and Mubarak Shah. Learning, detection and representation of multi-agent events in videos, 2007.
- [7] Henry A. Kautz, 'A formal theory of plan recognition and its implementation', in *Reasoning about Plans*, pp. 69–125. Morgan Kaufmann, (1991).
- [8] Lin Liao, Dieter Fox, and Henry Kautz, 'Extracting places and activities from gps traces using hierarchical conditional random fields', *The International Journal of Robotics Research*, **26**(1), 119–134, (2007).
- [9] Nam T. Nguyen, Hung H. Bui, Svetha Venkatesh, and Geoff West, 'Recognising and monitoring high-level behaviours in complex spatial environments', in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 620–625, (2003).
- [10] Nam T. Nguyen, Dinh Q. Phung, Svetha Venkatesh, and Hung Bui, 'Learning and detecting activities from movement trajectories using the hierarchical hidden markov models', in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 955–960, (2005).
- [11] N. M. Oliver, A. Garg, and E. Horvitz, 'Layered representations for learning and inferring office activity from multiple sensory channels', *Computer Vision and Image Understanding*, **1**(2), 163–180, (August 2002).
- [12] Lawrence R. Rabiner, 'A tutorial on hidden markov models and selected applications in speech recognition', in *Proceedings of the IEEE*, volume 77, pp. 257–286, San Francisco, CA, USA, (1989).
- [13] Gita Sukthankar and Katia Sycara, 'Robust recognition of physical team behaviors using spatio-temporal models', in *The fifth international joint conference on Autonomous agents and multiagent systems (AAMAS '06)*, pp. 638–645, (2006).