Learning Variable Length Markov Models of Behaviour

Aphrodite Galata, Neil Johnson and David Hogg

School of Computing

The University of Leeds

Leeds, LS2 9JT

United Kingdom

Telephone: +44 113 233 {6818, 5765}

Facsimile: +44 113 233 5468

Email: {afro, dch}@comp.leeds.ac.uk

Abstract

In recent years there has been an increased interest in the modelling and recognition of human activities involving highly structured and semantically rich behaviour such as dance, aerobics, and sign language. A novel approach is presented for automatically acquiring stochastic models of the high-level structure of an activity without the assumption of any prior knowledge. The process involves temporal segmentation into plausible atomic behaviour components and the use of variable length Markov models for the efficient representation of behaviours. Experimental results are presented which demonstrate the synthesis of realistic sample behaviours and the performance of models for long-term temporal prediction.

Keywords: modelling behaviour, behaviour prediction, behaviour synthesis, variable length Markov models, Markov models, N-grams, hidden Markov models, probabilistic finite state automata, statistical grammars, computer animation.

1 Introduction

In recent years, challenging problems such as human-computer interaction, automated visual surveillance and the realistic animation of human motion, have led to an increased interest in providing machines with the ability to learn and use models of human behaviour [15, 3, 13, 8]. Of particular interest is the modelling and recognition of human activities involving highly structured and semantically rich behaviour such as dance, aerobics, and sign language [6, 20, 21].

In this paper an activity is viewed as a sequence of primitive movements with a high-level structure controlling the temporal ordering. Others have used a similar approach to perceiving human activities. Bobick and Ivanov [4] used a context-free parsing mechanism together with HMMs [16] modelling low-level behaviour primitives for the recognition of activities. They used a hand-coded stochastic context-free grammar to represent *a priori* knowledge of the high-level structure of an activity. Bregler [5] proposed a framework for the probabilistic decomposition of human dynamics at different levels of abstraction, modelling complex gestures as successive phases of simple movements using an HMM. Similarly, Pentland and Liu [14] and Rittscher and Blake [17] used a set of dynamic models coupled together with a Markov chain representing long-term continuity constraints.

Unfortunately, HMMs do not encode high order temporal dependencies easily. Local optima are frequently encountered by iterative optimisation techniques when learning HMMs with many free parameters, and thus model topology and size are often highly constrained prior to training. We propose the use of variable length Markov models (VLMM) [18, 9] as a simple, yet powerful and efficient mechanism for capturing behavioural dependencies and constraints. Using a cross-entropy measure, VLMMs are able to locally optimise memory length within the model. This efficiently captures long-term temporal dependencies in some parts of behaviour and short-term dependencies elsewhere.

Two different methods are proposed for modelling behaviour using VLMMs at different temporal

scales. In the first, a VLMM is used to encode sequences of prototypical configurations of short-term structure and motion (typically spanning of the order of 20ms). In the second, a VLMM is used to encode sequences of atomic behaviours segmented automatically from training data and representing primitive actions (typically lasting of the order of a second) such as raising an arm. The advantage of using a VLMM at this higher level is that it facilitates a more powerful encoding of temporal dependencies.

We show how both kinds of model can be learned from extended video sequences depicting target behaviours. Finally, we demonstrate the use of the learnt behaviour models in two applications. In the first, we show how they may be used for animation of human activity; producing statistically accurate variations on an action without requiring operator intervention. In the second, we demonstrate their use in prediction, aimed at improving the reliability of object tracking or anticipating someone's actions in the immediate future.

2 Augmented configuration space

Behaviours may be viewed as smooth trajectories within an appropriate feature space. Our modelling framework is based on an *augmented configuration space* which describes both d-dimensional object configuration $\mathbf{C}_t$ and its first derivative $\dot{\mathbf{C}}_t$. The inclusion of derivatives helps resolve ambiguities in configuration space. Moreover, it facilitates the use of models in performing generative tasks (see section 3.2).

The first stage of the modelling process involves the acquisition of sequences, $\mathcal{F}_j$, of regularly sampled augmented configuration vectors, $\mathbf{F}_t \in [0,1]^{2d}$. Each sequence describes the temporal evolution of a behaviour:

$$\mathcal{F} = \{\mathbf{F}_0,\ \mathbf{F}_1,\ \ldots,\ \mathbf{F}_m\}\,, \tag{1}$$

where

$$\mathbf{F}_t = \left(\mathbf{C}_t,\ \lambda\dot{\mathbf{C}}_t\right), \tag{2}$$

and λ balances the contribution of derivatives when using the Euclidean distance as a dissimilarity measure.

In order to generate a discrete representation of behaviours, we replace each vector $\mathbf{F}_t$ by the nearest (in a Euclidean sense) prototype from a finite set $\mathcal{P} = \{\mathbf{p}_0,\ \mathbf{p}_1,\ \ldots,\ \mathbf{p}_n\}$ of prototypical augmented configurations. Multiple occurances of the same prototype are replaced by a single occurance. Prototypes are derived using robust vector quantisation (see Johnson and Hogg [13] for details). Each behaviour is therefore represented by a sequence of prototypes:

$$\{\mathbf{p}_{i_0},\ \mathbf{p}_{i_1},\ \ldots,\ \mathbf{p}_{i_\mu}\} \tag{3}$$

where $\mu \leq m$.

2.1 Experimental data

To demonstrate the generality of our approach, two data sets with entirely different configuration spaces are used for the experiments in Section 3 and 4.

2.1.1 Data set 1: 2–D Contour tracking

For the first data set, individuals performing exercise routines were tracked using a simple contour tracker [12, 1]. Object configuration is represented by the n control points of a closed uniform B-spline approximating the silhouette boundary. Control points are evenly spaced around the silhouette and ordered relative to a consistent point of reference. The method for the location of this reference point has been enchanced from [1] to allow the top of an individual's head to be more accurately located. This enchancement involves local adjustment of the reference point such that it coincides with the locally

highest part of the silhoutte boundary. Figure 1 illustrates this shape representation, showing a sample silhouette boundary, (a)(ii), alongside the corresponding video image, (a)(i).

Spline control points are transformed from image coordinates into an object centred coordinate system and normalised such that each component of the transformed control points lies in the interval $[0, 1]$. The evolving silhouette boundary is thus represented by configuration vectors $\mathbf{C}_t \in [0, 1]^{2n}$ (i.e. $d = 2n$):

$$\mathbf{C}_t = (x_1(t),\ y_1(t),\ x_2(t),\ y_2(t),\ \ldots,\ x_n(t),\ y_n(t))\,. \tag{4}$$

Training data was generated from a 40 second sequence of an individual performing an exercise routine, sampled at 25 frames per second. This exercise routine comprises two exercises, each repeated four times and followed by four repetitions of a sub-exercise (see Figure 1(b)). Splines with 32 control points were used, resulting in a 128-dimensional augmented configuration space ($2 \times 2 \times 32$). Using a scaling factor of $\lambda = 10$, a set of 71 prototypical augmented configuration vectors were learnt from this data set.

2.1.2 Data set 2: 3–D Motion capture

For the second data set, we used a commercially available system (Camera: ProReflex, Software: MacReflex, Qualisis 97) to provide the 3-D locations of markers attached to the body of individuals performing exercise routines. 3-D Motion capture data were recorded from 13 points on the human body at a rate of 50 frames per second.

Integration of data from different training sequences is enabled by transforming 3-D data points into object centred coordinates and normalising such that each component of the transformed points lies in the interval $[0, 1]$. Object behaviour is thus represented by ordered sequences of augmented configuration vectors $\mathbf{F}_t \in [0, 1]^d$, where $d = 72$ ($2 \times 3 \times 13$).

Training data was generated from eight sequences (of approximate duration 25 seconds each) of an individual perfoming the exercise routine shown in Figure 2. Using a scaling factor of $\lambda = 30$, a set of 87 prototypical augmented configuration vectors were learnt from this data set.

3 Modelling behaviour using variable length Markov models

We are interested in building models of behaviour which are able to support both recognition and generative capabilities such as the prediction of future behaviours or the synthesis of realistic sample behaviours. We achieve this by using variable memory length Markov models (VLMMs) [9] to encode the sequences of prototype vectors corresponding to observed behaviours in the augmented configuration space.

3.1 Variable length Markov models

Variable length Markov models deal with a class of random processes in which the memory length varies, in contrast to an n-th order Markov model for which the memory length is fixed. They have been used successfully for text compression [7, 2] and more recently in language modelling to improve the accuracy of speech and handwriting recognisers [18, 9, 10]. Their advantage over a fixed memory Markov model is the ability to locally optimise the length of memory required for prediction. This results in a more flexible and efficient representation which is particularly attractive in cases where we need to capture higher-order temporal dependencies in some parts of the behaviour and lower-order dependencies elsewhere.

Assume w is a string of tokens used as a memory to predict the next token a' according to an estimate $\hat{\mathrm{P}}(a'|w)$ of $\mathrm{P}(a'|w)$. The main idea behind the variable length modelling method is that if the output probability $\hat{\mathrm{P}}(a'|aw)$ that predicts the next token a' is significantly different from $\hat{\mathrm{P}}(a'|w)$, then the longer memory aw may be a better predictor than w. A weighted Kullback-Leibler divergence [18] is used to measure the additional information that is gained by using the longer memory aw for prediction instead of the shorter memory w:

$$\Delta H(aw, w) = \hat{\mathrm{P}}(aw) \sum_{a'} \hat{\mathrm{P}}(a'|aw) \log \frac{\hat{\mathrm{P}}(a'|aw)}{\hat{\mathrm{P}}(a'|w)}. \tag{5}$$

If $\Delta H(aw, w)$ exceeds a given threshold ε, then the longer memory aw is used, otherwise the shorter

memory w is considered sufficient for prediction.

The transition probabilities and priors are derived from estimates of $\mathrm{P}(a_n|a_1a_2\ldots a_{n-1})$ and $\mathrm{P}(a_1a_2\ldots a_n)$ calculated for various values of n ($n = 1, 2, \ldots, N$). The estimates are given by:

$$\hat{\mathrm{P}}(a_n|a_1a_2\ldots a_{n-1}) = \frac{v(a_1a_2\ldots a_{n-1}a_n)}{v(a_1a_2\ldots a_{n-1})}, \tag{6}$$

and

$$\hat{\mathrm{P}}(a_1a_2\ldots a_n) = \frac{v(a_1a_2\ldots a_n)}{v_0}, \tag{7}$$

where $v(a_1a_2\ldots a_n)$ is the number of times the string of tokens $a_1a_2\ldots a_n$ appears in the training data and v_0 is the total length of the training sequences.

The training algorithm involves building a *prefix tree* [9] where each node corresponds to a string up to a predetermined length N. The transition frequencies are counted by traversing the tree structure repeatedly with strings of length N where the strings are generated by sliding a window of fixed length N along a training sequence of tokens. Transition probabilities are computed using equation 6 and a pruning procedure is then applied while the prefix tree is converted to a *prediction suffix tree* [18]. For each node n_p in the prefix tree, a corresponding node n_s in the suffix tree is created if and only if the Kullback-Leibler divergence between the probability distribution at n_p and the probability distribution at its ancestor node in the prefix tree is larger than threshold ε. Finally, the suffix tree is converted to an automaton representing the trained VLMM. A more detailed description on building and training variable length Markov models is given by Ron *et al.* [18].

Thus, a VLMM is equivalent to a *Probabilistic Finite State Automaton* (PFSA) represented by $\mathcal{M} = (Q, \Sigma, \tau, \gamma, \pi)$ where Σ is a finite *alphabet* –the set of tokens– and Q is a finite set of model states. Each state corresponds to a token string of length at most N, ($N \geq 0$), representing the *memory* for a conditional transition of the VLMM. The *transition function* is $\tau : Q \times \Sigma \to Q$ and $\gamma : Q \times \Sigma \to [0, 1]$ is the *output probability function* representing the memory conditioned probabilities of the next *token*

$a \in \Sigma$. Finally, $\pi : Q \rightarrow [0,1]$ is the probability distribution over the *start states*. The functions γ and π are such that for every $q \in Q$, $\sum_{a \in \Sigma} \gamma(q,a) = 1$ and $\sum_{q \in Q} \pi(q) = 1$.

3.2 Modelling behaviour using a VLMM over prototypes

In our first approach, temporal dependencies in behaviour are learned by using a VLMM to capture the memory conditioned probabilities of transitions between prototypes. Initially, the training sequences are converted into sequences of prototypes by observing the closest prototype, in a *nearest neighbour* sense, to the current training vector at each time instant. Only transitions between different prototypes are considered for training. The output sequences are then used to train a VLMM represented by the PFSA $\mathcal{M}_p$.

3.2.1 Behaviour Generation

The trained model $\mathcal{M}_p$ represents the learnt behaviour model and has generative capabilities. Behaviour generation is achieved by traversing the PFSA, selecting either the most likely transition (maximum likelihood behaviour generation) or sampling from the transition distribution (stochastic behaviour generation) at each state, and emitting the corresponding prototype vectors. This results in an ordered set $\mathcal{G}$ of prototype vectors $\mathbf{p}_{q_r}$ which are the output of the transitions $q_{r+1} = \tau_b(q_r, \mathbf{p}_{q_r})$ between states q_r, q_{r+1}.

The time interval between generated prototypes is initially unspecified (due to the removal of repeated prototypes–see Section 2) and in order to generate an output sequence of vectors in the augmented configuration space at video frame rates, an interpolant of $\mathcal{G}$ must be sampled. Since non-linear changes may occur between prototypes separated by large time intervals, a (cubic) Hermite interpolation is used. Assuming constant acceleration, the time interval δ_r between successive prototype vectors can be

approximated by:

$$\delta_r = 2\frac{|\mathbf{C}_{r+1} - \mathbf{C}_r|}{|\dot{\mathbf{C}}_{r+1}| + |\dot{\mathbf{C}}_r|}. \tag{8}$$

The Hermite interpolant is defined by the endpoints $\mathbf{C}_r$ and $\mathbf{C}_{r+1}$ and tangent vectors $\dot{\mathbf{C}}_r$ and $\dot{\mathbf{C}}_{r+1}$ (scaled by δ_r). Using the approximate time intervals and interpolants between successive vectors, a temporally regular extrapolation can be produced by sampling at data frame rate.

3.2.2 Prediction of future behaviour

To use the model $\mathcal{M}_p$ for behaviour prediction, it is first necessary to locate the current model state. Since model states may encode a history of previous behaviour, the model is initially used in a recognition mode, accepting successive prototypes representing observed behaviour and making the corresponding state transitions. Having located the current model state, prediction of future behaviour can be achieved using the model either as a stochastic or a maximum likelihood behaviour generator.

A VLMM needs to be able to handle the problem of *unseen events* – cases where token sequences which might have not appeared previously in the training corpus, appear during the recognition process. Therefore, if the model $\mathcal{M}_p$ is presented with a prototype $\mathbf{p}_i \in \mathcal{P}$ while in a state which emits this prototype with probability zero, we return to the initial model state, lose all the previous memory and predict $\mathbf{p}_i$ with the prior probability $\hat{\mathrm{P}}(\mathbf{p}_i)$. This *backing-off* method is simple but effective, although other more complex methods of handling unseen events could be employed [11].

3.2.3 Assessing predictor performance

For each prediction model a set of mean errors are calculated to quantify the performance in predicting the value of the output vector in configuration space on each future time instant:

$$E_T = \frac{\sum_{j=1}^{n} \left\| \mathbf{C}_{t+T}^{j} - \mathbf{C}_{t+T}^{*} \right\|}{n}, \tag{9}$$

where n is the total number of trials carried out over all test sequences. The error in predicting forward by T time steps is averaged over predictions generated on every frame of every test sequence and $\mathbf{C}^*_{t+T}$ denotes the *ground truth* vector in configuration space at time $t+T$ as given by the test data. Errors are only updated if both a prediction and the ground truth exist for the particular T.

The mean performance of the models should represent a probabilistic weighting of the errors given by all possible predictions. In general, it is not possible to enumerate the entire set of possible predictions from a particular model state due to the possibility of cycles within the transition structure. Instead, mean performance is calculated using Monte Carlo simulation, generating a large number of stochastic predictions on each frame and allowing their relative frequency to provide probabilistic weighting within the calculation of mean errors. For the experiments presented in this paper, 50 stochastic predictions were generated on each frame.

3.2.4 Experiments

Using the training data and the set of prototypes described in Section 2.1, variable length Markov models over prototypes were used to encode behavioural dependencies. Two different values of the maximum allowed memory length N were used for these models.

Data set 1: 2-D Contour tracking

The learnt set of 71 prototypes (see Section 2.1.1) was used as an alphabet and variable length Markov models were trained using two maximum memory lengths of $N = 5$ and $N = 20$, and a threshold on the Kullback-Leibler divergence of $\epsilon = 0.0001$. The resulting models had 117 and 290 states respectively. Figure 3 shows the histogram of state memory length frequency within the learnt VLMM for (a) $N = 5$, and (b) $N = 20$ respectively.

A test sequence of an individual performing an exercise routine was used to demonstrate predictor

performance. The test sequence was composed of the same two exercises and sub-exercises as those in the training data, but in this case each exercise and sub-exercise was repeated three times. Figure 4(a) demonstrates predictor performance of the variable length behaviour model for the two different values of N and is compared with the performance of a first order Markov model. Each plot in the graph demonstrates mean predictor performance over a range $t + T$ $(1 \leq T \leq 70)$ of future time instants (≈ 3 sec.), averaged over all stochastic predictions for all frames in the test data set.

Data set 2: 3-D Motion capture

VLMMs over the set of 87 prototypes (see Section 2.1.2) were also trained using memories of maximum length of 5 and 20 with a threshold $\varepsilon = 0.0001$. The resulting models have 112 and 226 states respectively.

Two different test sequences of an individual performing an exercise routine composed of the same three exercises and sub-exercises as those in the training data were used to demonstrate predictor performance. The test sequences were approximately 25 seconds long. As before, figure 4(b) demonstrates predictor performance. Each plot in the graph demonstrates mean predictor performance up to about 3 seconds in the future, averaged over all stochastic predictions for all frames in both test data sequences.

As can be seen from the prediction graphs in figures 4(a) and (b), substantially better results are obtained using VLMMs in comparison to a first order Markov model. Learning temporal behaviour dependencies at the prototype level using a variable length Markov model results in an efficient behaviour model representation that captures accurately local behaviour dependencies and has good generative capabilities. However, in order to capture the 'syntactic' structure of an activity over a longer time-scale, simultaneously with the short term structure necessary for animation, a hierarchical approach in behaviour modelling as described in the next section can yield better results.

4 Learning structured behaviour models

Motion in human activities has different characteristics at different time scales, usually carrying syntactic and semantic information at larger temporal scales. Capturing the variability of behaviour at different temporal scales could be facilitated by using multiple memory mechanisms, thus providing a more powerful means of modelling structured and semantically rich behaviours of human activities.

In this section a hierarchical memory mechanism is proposed using a VLMM encoding sequences of atomic behaviours, each of which have their own stochastic micro-structure. Using such a mechanism, it is possible to capture the larger scale temporal dependencies and therefore infer possible high level syntactic or semantic information about the studied behaviour.

4.1 Learning atomic behaviours

Temporal segmentation into atomic behaviours involves identifying suitable break points at which to partition an activity. Given the physical constraints posed by the human body, any change in the type of human movement usually causes dips in velocity. We wish to exploit this by using minima in the magnitude of configuration change as clues for performing semantic temporal segmentation.

Behaviour, as described in Section 2, is represented by a sequence of prototypes in the augmented configuration space, describing the temporal evolution of behaviour. We identify those prototypes for which the magnitude of the first derivative $\dot{\mathbf{C}}_t$ is a local minimum and below a fixed threshold (chosen by inspection). We call this the set of *key prototypes* $\mathcal{K}$:

$$\mathcal{K} = \{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_{k-1}, \mathbf{k}_k\} \subset \mathcal{P}. \tag{10}$$

These key prototypes are then utilised to facilitate the learning of larger scale temporal dependencies in behaviour. Ideally, the segmentation points might be derived from a global maximisation of likelihood over the training data. Such an approach is subject of future research.

The set of key prototypes uniquely define a set of *atomic behaviours* that start and end with these prototypes. Specifically, the atomic behaviour α_{ij} represents the range of behaviour observed between key prototypes $\mathbf{k}_i$ and $\mathbf{k}_j$ for $i \neq j$. Each atomic behaviour component α_{ij} comprises a set of m *template sequences* α_{ij}^l, $1 \leq l \leq m$, where m is, in general, different for each atomic behaviour component. Each such template is an ordered set of n augmented configuration prototypes

$$\alpha_{ij}^l = \{\mathbf{p}_{\iota_1},\ \mathbf{p}_{\iota_2},\ \ldots,\ \mathbf{p}_{\iota_n}\}, \tag{11}$$

where n is, in general, different for each template sequence.

Template sequences are generated from the training corpus via a process of sequence clustering and merging. Initially, the set of all training sub-sequences spanning the transition between prototypes $\mathbf{k}_i$ and $\mathbf{k}_j$ is acquired. This set is then partitioned into a number of clusters of self-similar sequences using dynamic time warping [19] to assess sequence similarity. Finally, a single template sequence is generated from each cluster via a process of sequence re-sampling, averaging, and re-quantisation.

The relative probability $\mathrm{P}(\alpha_{ij}^l)$, $\sum_l \mathrm{P}(\alpha_{ij}^l) = 1$, of each template sequence within the atomic behaviour component α_{ij} is derived from the training corpus by observing the relative frequency with which the templates are matched.

4.2 Inferring a higher-level behaviour grammar

Temporal segmentation into atomic behaviours enables the learning of a higher level behaviour model that efficiently captures temporal ordering and constraints between constituent atomic behaviours. This is achieved using a VLMM to capture the memory conditioned probabilities of transitions between atomic behaviours. For convenience, the VLMM $\mathcal{M}_k = (Q_k, \mathcal{K}, \tau_k, \gamma_k, \pi_k)$ is trained using the set of key prototypes as an alphabet. Although key prototypes are used as the alphabet, the memory conditioned probabilities of transitions between atomic behaviours are implicit within the model. Suppose $\mathrm{P}_{\mathcal{M}_k}(\mathbf{k}_j | s\mathbf{k}_i)$

denotes the probability of observing key prototype $\mathbf{k}_j$ conditioned on the observation of history $s\mathbf{k}_i$, where s denotes a key prototype history. Since α_{ij} is the atomic behaviour component representing the transition between $\mathbf{k}_i$ and $\mathbf{k}_j$, clearly $\mathrm{P}_{\mathcal{M}_k}(\alpha_{ij}|s\mathbf{k}_i) = \mathrm{P}_{\mathcal{M}_k}(\mathbf{k}_j|s\mathbf{k}_i)$. Within such a behaviour model, the probability $\mathrm{P}_{\mathcal{M}_k}(\alpha_{ij}^l|s\mathbf{k}_i)$ of observing template sequence α_{ij}^l, conditioned on the observation of a key prototype history $s\mathbf{k}_i$, is given by:

$$\mathrm{P}_{\mathcal{M}_k}(\alpha_{ij}^l|s\mathbf{k}_i) = \mathrm{P}(\alpha_{ij}^l)\mathrm{P}_{\mathcal{M}_k}(\alpha_{ij}|s\mathbf{k}_i). \tag{12}$$

Figure 5 illustrates part of a sample VLMM encoding the high-level structure of a behaviour. Also illustrated is the sub-model of an atomic behaviour component defined between two key prototypes that is implicitly built within the high-level model.

4.3 Behaviour generation

The learnt structured behaviour model has stronger generative capabilities compared with the model in Section 3.2.1. This is due to the fact that temporal behaviour dependencies are encoded at a higher level offering a more powerful, longer duration memory mechanism.

Behaviour generation is achieved by traversing the model's automaton $\mathcal{M}_k$, generating a key prototype at each step and replacing each $\mathbf{k}_i\mathbf{k}_j$ subsequence with a $\mathbf{k}_i\alpha_{ij}^l\mathbf{k}_j$ subsequence. The choice of template α_{ij}^l representing the atomic behaviour α_{ij} is achieved by either choosing the template that maximises equation 12 or sampling from the set of possible templates α_{ij}^l.

Entirely hypothetical sequences can be generated using the start state distribution π_k to select an initial model state. The selection of the start state is based on either sampling from the start state distribution or identifying the most probable state. The start state distribution is approximated by the relative frequency of starting at a particular state of the VLMM in the training data.

Figure 6 illustrates stochastic synthesis of sample behaviour using the learnt behaviour model for the

3-D motion capture example (see section 4.5). An entirely hypothetical excersise routine sequence has been generated and is used to animate a virtual humanoid using the VRML modelling language.

4.4 Prediction of future behaviour

In order to use the model $\mathcal{M}_k$ for behaviour prediction, it is necessary not only to locate the current model state (as in section 3.2.2), but also to identify the atomic behaviour template currently being observed and to locate the current position within this template. Once the atomic behaviour has been found, the subsequent model state is implicitly identified.

Suppose that, at time t, it has been established that the current state of the model $\mathcal{M}_k$ is q_c, having memory $s\mathbf{k}_i$, and that the prototype sequence $O_t = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{t-1}, \mathbf{o}_t\}$ has been observed since the last observed key $\mathbf{k}_i$. In order to select the subsequent model state, it is necessary to identify the atomic behaviour template currently being traversed.

A Bayesian approach is taken where the estimated transition probabilities of the learned model $\mathcal{M}_k$ are used as *priors*. The posterior probability that atomic behaviour template α_{ij}^l represents the observation sequence at time t, taking into account the history of the high-level behaviour model, is approximated by:

$$\mathrm{P}(\alpha_{ij}^l | O_t, s\mathbf{k}_i) \propto \mathrm{P}(O_t | \alpha_{ij}^l) \mathrm{P}_{\mathcal{M}_k}(\alpha_{ij}^l | s\mathbf{k}_i), \tag{13}$$

where $\mathrm{P}_{\mathcal{M}_k}(\alpha_{ij}^l | s\mathbf{k}_i)$ is given by equation 12 and $\mathrm{P}(O|\alpha_{ij}^l)$ is the likelihood of template α_{ij}^l giving rise to the current observation sequence.

All the atomic behaviour template sequences from all the possible atomic behaviours defined from the last observed key prototype $\mathbf{k}_i$ to any other key prototype $\mathbf{k}_q$ are considered. For each such template sequence α_{iq}^r, a position $\xi \leq n$ is found such that the distance or the cost to align $\{\mathbf{p}_{\iota_1}, \mathbf{p}_{\iota_2}, \ldots, \mathbf{p}_{\iota_\xi}\}$ with $\{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t\}$ is minimised using dynamic time warping. This minimum cost is denoted by $\mathrm{c}_t(i, q, r)$. Utilising the cost function, it is possible to approximate the likelihood $\mathrm{P}(O_t|\alpha_{ij}^l)$ with the relative prob-

ability of atomic behaviour template α_{ij}^{l} giving rise to the observation sequence O_t:

$$\mathrm{P}(O_t|\alpha_{ij}^{l}) = 1 - \frac{\mathrm{c}_t(i,j,l)}{\sum_{q,r}\mathrm{c}_t(i,q,r)}. \tag{14}$$

Selecting the atomic behaviour template for which equation 13 is maximised identifies the subsequent key prototype $\mathbf{k}_j$ and thus the subsequent state of the VLMM.

Unseen events are handled by detecting cases in which the maximum probability is less than a threshold θ. In these cases, we loose all previous history of the high-level behaviour model and predict the next key prototype using only the likelihood function given by equation 14.

Having located the next state q_{c+1} of the VLMM $\mathcal{M}_k$ and the position ξ within the atomic behaviour template currently being traversed, generation of future behaviour is possible. The generated behaviour comprises the remaining template behaviour sequence $\{\mathbf{o}_{\xi+1}, \mathbf{o}_{\xi+2}, \ldots, \mathbf{o}_{t-1}, \mathbf{o}_t\}$ and the behaviour generated from the state q_{c+1} using the model $\mathcal{M}_k$ either as a stochastic or a maximum likelihood behaviour generator as described in the previous section.

Figure 7 illustrates an example of maximum likelihood future behaviour extrapolation at selected time instants during the evolution of an exercise routine. In each figure, recent behaviour is illustrated by a set of filled contours, the shade of which indicates recency, the lightest being the current shape. The first 12 frames of each extrapolation are illustrated.

4.5 Experiments

Using the training data and the set of prototypes described in Section 2.1, structured behaviour models have been learned with VLMMs capturing the structure of activity at a higher level using atomic behaviours as an alphabet.

Data set 1: 2-D Contour tracking

From the set of 71 prototypes (section 2.1.1), 5 prototypes have been identified as key prototypes and 8 atomic behaviours representing the range of behaviour between key prototypes have been learned. VLMMs were trained using memories of maximum length N equal to 4, 6 and 8 with a threshold $\varepsilon = 0.0001$, resulting in models with 22, 34 and 45 states respectively. Figure 8 shows the histogram of state memory length frequency within the learnt VLMM for (a) $N = 4$, (b) $N = 6$, and (c) $N = 8$ respectively.

Figure 9 demonstrates the generation of sample behaviours using the learnt structured behaviour models (with maximum memory (a) N=1, (b) N=4, and (c) N=8 respectively). Stochastic extrapolation and selection of initial states were used to create entirely synthetic exercise routine sequences. In each figure, sequences are illustrated by a set of filled contours progressing in a top-to-bottom order. For illustrative purposes every second frame of the first 700 frames is displayed. With N=1 (equivalent to a first order Markov model), the model does not capture longer-term temporal constraints between atomic behaviours as can be seen by the random order in which the separate exercises and sub-exercises are generated. In figures 9(b) and 9(c), the encoding of longer-term temporal dependencies is illustrated by the correct progression from one exercise or sub-exercise to the next.

The test sequence from section 3.2.4 is again used here to assess prediction performance. Figure 10(a) illustrates predictor performance using the learned structured behaviour model for $N = 6$. The performance of the behaviour model described in section 3.2 is also illustrated here for comparison.

Data set 2: 3-D Motion capture

From the set of 87 prototypes (section 2.1.2), 5 prototypes have been identified as key prototypes and 8 atomic behaviours representing the range of behaviour between key prototypes have been learned. VLMMs were trained using memories of maximum length N equal to 2 and 4 with a threshold $\varepsilon = 0.0001$, resulting in models with 9 and 12 states respectively. Figure 6 illustrates the stochastic generation of

sample behaviour of the learnt structured behaviour model for $N = 4$.

The two test sequences from section 3.2.4 are again used here to assess prediction performance. Figure 10(b) illustrates predictor performance using the learned structured behaviour model for $N = 4$. The performance of the behaviour model described in section 3.2 is also illustrated here for comparison.

As can be seen from the prediction graphs in figures 10(a) and 10(b), the structured behaviour models consistently give better prediction results, demonstrating the increased ability of the model to encode the complex, long-term temporal dependencies.

5 Conclusions

Two novel methods have been proposed for the automatic acquisition of statistical models for structured and semantically rich behaviours. The use of variable length Markov models provides an efficient mechanism for learning complex behavioural dependencies and constraints.

Capturing behaviour at a low level using a VLMM with prototypical configurations as an alphabet, results in a model that accurately encodes local behaviour dependencies. However, such a model is unable to capture dependencies at multiple temporal scales. Using a VLMM at a higher level of abstraction, with constituent atomic behaviours as an alphabet, it is possible to automatically infer a stochastic model of the high-level structure of a behaviour. The learned structured behaviour model encodes behavioural dependencies at two temporal scales thus providing a more powerful model.

Both models have good generative capabilitites and can be utilised for behaviour recognition, for the automatic synthesis of realistic object behaviours and for improving the robustness and efficiency of object tracking systems. Extention of the proposed two level hierarchy to multiple levels could be envisaged mirroring behavioural hierarchies occuring in nature. This could be achieved using a hierarchy of VLMMs to capture multiple levels of motion abstraction.

Acknowledgments

The authors would like to thank Christian Babski of the Computer Graphics Lab, Swiss federal Institute of Technology for providing 'Baxter', a standard H-Anim1.1 VRML model for humanoids, and the Centre for Studies in Physical Education and Sports Science at University of Leeds for providing access to their 3-D motion capture facilities.

References

[1] A. Baumberg and D. Hogg. Learning Flexible Models from Image Sequences. In *European Conference on Computer Vision*, volume 1, pages 299–308, 1994.

[2] T. Bell, J. Cleary, and I. Witten. *Text Compression*. Prentice Hall, 1990.

[3] A. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. In *Philosophical Transactions Royal Society London*, 1997.

[4] A.F. Bobick and Y.A. Ivanov. Action Recognition using Probabilistic Parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 196–202, 1998.

[5] C. Bregler. Learning and Recognising Human Dynamics in Video Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[6] L. Campbell and A. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *International Conference on Computer Vision*, pages 624–630, June 1995.

[7] G. Cormack and R. Horspool. Data Compression using Dynamic Markov Modelling. *Computer Journal*, 30(6):541–550, 1987.

[8] D.M. Gavrila. The Visual Analysis of Human Movement: A Survey. *CVIU*, 73(1):82–98, 1999.

[9] I. Guyon and F. Pereira. Design of a Linguistic Postprocessor using Variable Memory Length Markov Models. In *International Conference on Document Analysis and Recognition*, pages 454–457, 1995.

[10] J. Hu, W. Turin, and M Brown. Language Modelling using Stochastic Automata with Variable Length Contexts. *Computer Speech and Language*, 11(1):1–16, 1997.

[11] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[12] N. Johnson, A. Galata, and D. Hogg. The Acquisition and Use of Interaction Behaviour Models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 866–871, 1998.

[13] N. Johnson and D. Hogg. Learning the Distribution of Object Trajectories for Event Recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

[14] A. Pentland and A. Liu. Modeling and Prediction of Human Behaviour. *Neural Computation*, 11:229–242, 1999.

[15] A. P. Pentland. Machine Understanding of Human Motion. Technical Report 350, MIT Media Laboratory Perceptual Computing Section, 1995.

[16] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[17] J. Rittscher and A. Blake. Classification of Human Body Motion. In *International Conference on Computer Vision*, pages 634–639, 1999.

[18] D. Ron, S. Singer, and N. Tishby. The Power of Amnesia. In *Advances in Neural Information Processing Systems*, volume 6, pages 176–183. Morgan Kauffmann, 1994.

[19] H Sakoe and S. Chiba. Dynamic Programming optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 26:623–625, 1980.

[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] C. Vogler and D. Metaxas. ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis. In *International Conference on Computer Vision*, pages 363–369, 1998.

List of figures

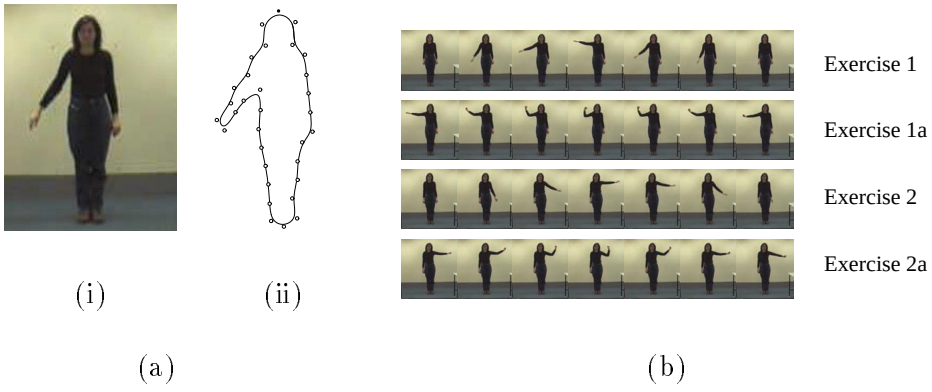


Figure 1: (a) Spline-based shape representation (b) Structure of exercise routine for the contour tracking example.

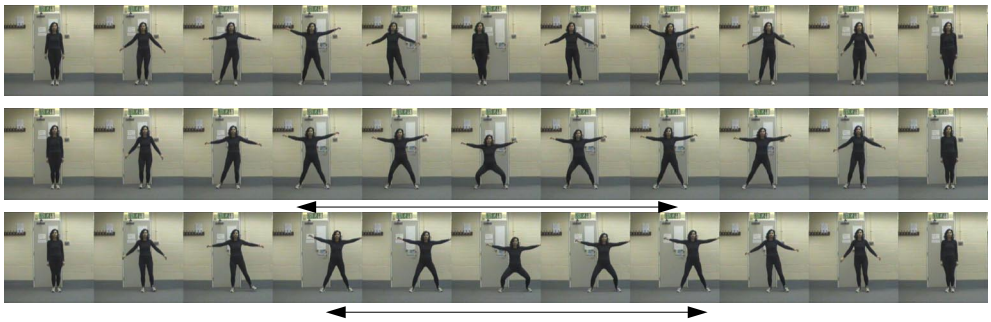
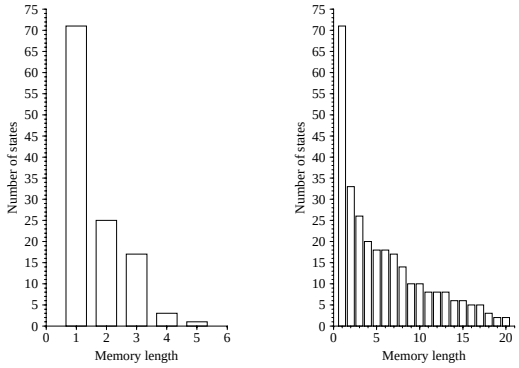


Figure 2: Structure of the exercise routine for the 3-D motion capture example: the exercise comprises three exercises with the first exercise repeated twice. The sub-exercise which occurs in the last two exercises is identified with arrows and is repeated twice within each exercise.



(a) (b)

Figure 3: Frequency histograms of state memory lengths within learnt VLMMs

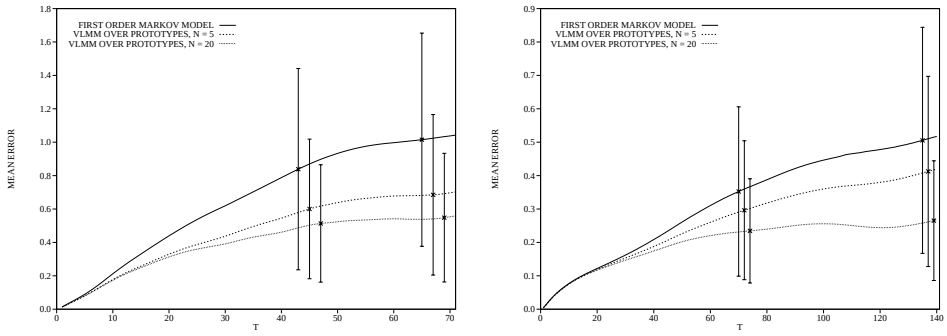


Figure 4: Comparing variable length Markov models over prototypes in the augmented configuration space of (a) 2-D exercise profiles and (b) 3-D motion capture data. The error bars show mean error $\pm$ mean absolute deviation.

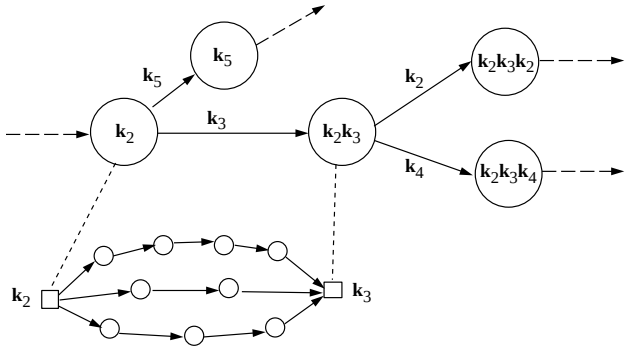


Figure 5: Illustrating a hierarchical behaviour model, showing states of the PFSA corresponding to the memories of the VLMM and three alternative templates for a single embedded atomic movement between $\mathbf{k}_2$ and $\mathbf{k}_3$

Figure 6: Animation of synthesized behaviour

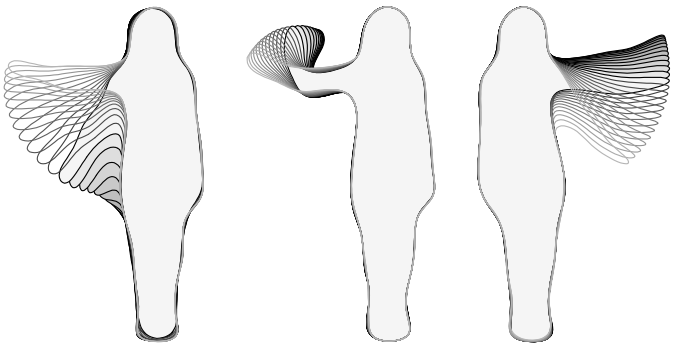
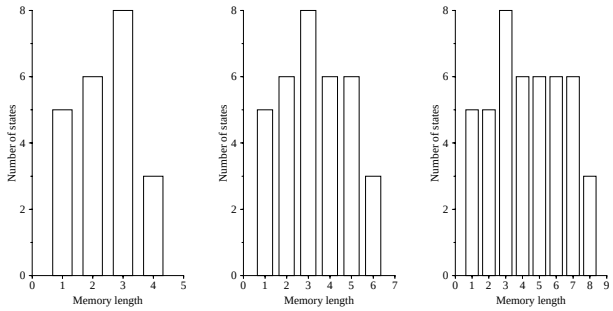
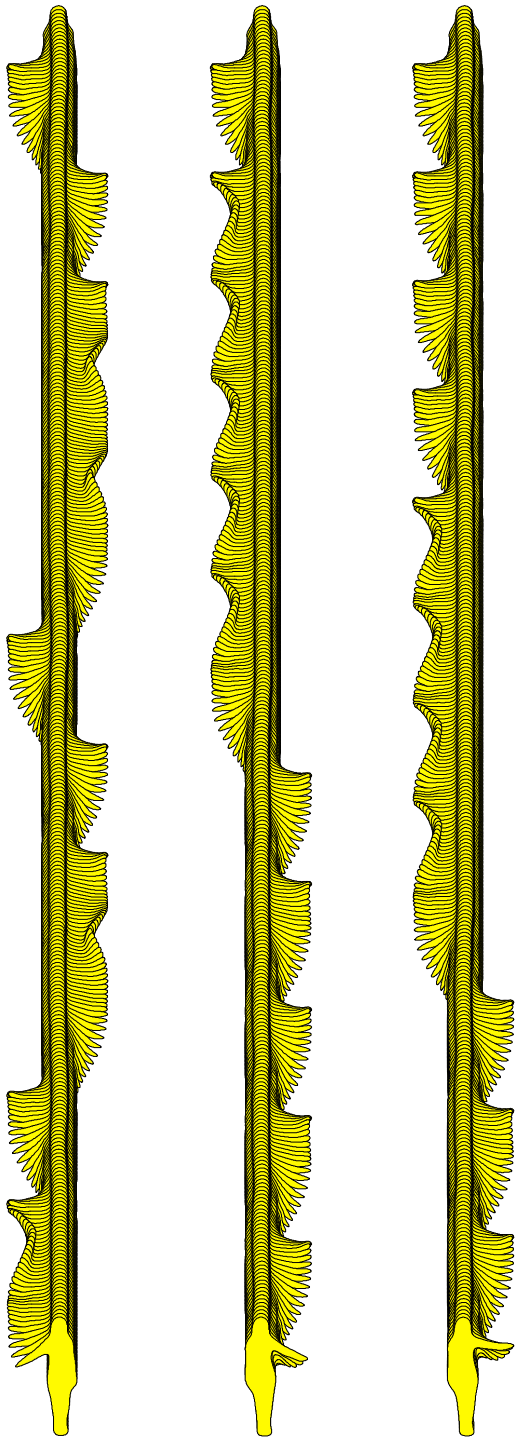


Figure 7: Behaviour extrapolation results.



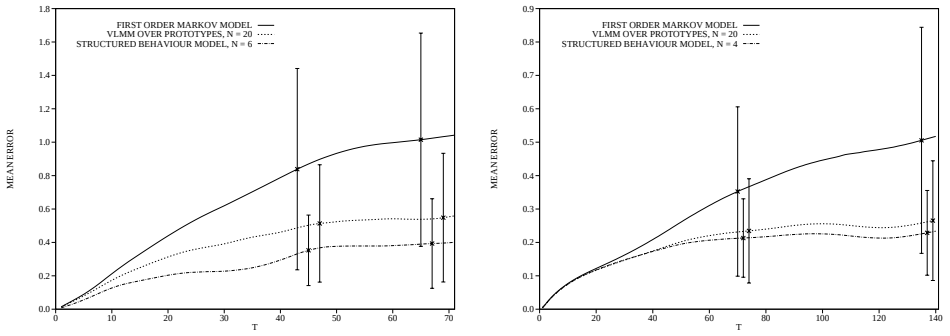
(a) (b) (c)

Figure 8: Frequency histograms of state memory lengths within learnt VLMMs



(a) $N = 1$ (b) $N = 4$ (c) $N = 8$

Figure 9: Sample exercise routine sequences



(a) (b)

Figure 10: Mean predictor permformance in the augmented configuration space of (a) 2-D exercise profiles and (b) 3-D motion capture data. The error bars show mean error ± mean absolute deviation.