Hindawi Publishing Corporation
International Journal of Computer Games Technology
Volume 2008, Article ID 751268, 7 pages
doi:10.1155/2008/751268

Research Article

Activity Classification for Interactive Game Interfaces

John Darby, Baihua Li, and Nick Costen

Department of Computing and Mathematics, The Manchester Metropolitan University, John Dalton Building, Chester Street, Manchester M1 5GD, UK

Correspondence should be addressed to John Darby, j.darby@mmu.ac.uk

Received 28 September 2007; Accepted 13 December 2007

Recommended by Kok Wai Wong

We present a technique for modeling and recognising human activity from moving light displays using hidden Markov models. We extract a small number of joint angles at each frame to form a feature vector. Continuous hidden Markov models are then trained with the resulting time series, one for each of a variety of human activity, using the Baum-Welch algorithm. Motion classification is then attempted by evaluation of the forward variable for each model using previously unseen test data. Experimental results based on real-world human motion capture data demonstrate the performance of the algorithm and some degree of robustness to data noise and human motion irregularity. This technique has potential applications in activity classification for gesture-based game interfaces and character animation.

1. INTRODUCTION

The interpretation of human motion is a fundamental task in computer vision. It has received much attention in recent years with wide applications in surveillance, human computer interaction, and the entertainment industry [1]. In vision-based interfaces for video games, such as that in [2] by Decathlete, a player's gestures and activities are used as commands for game control instead of pressing buttons on a keyboard or moving a mouse. In this case, the player's movements, embedded in video images, must be detected, parameterised, and recognised with a sufficient level of accuracy to allow interaction with an intelligent agent.

On the other hand, generating realistic human motion remains an open problem in the game industry. Traditional key-framing methods are extremely labour intensive requiring the manual specification of key poses at specific frames. Physical simulation seems to be more realistic than key-framing, but due to the difficulty of modelling the underlying control mechanism, instabilities, and high computation cost, physics-based animation has not been used with much success. Recently, performance-based animation has received much interest [3]. Among these techniques, marker-based or markerless video-driven animation has shown great potential [4, 5]. Low-level features, such as key-point positions or joint angles, are used to describe full-body movements. MPEG-4, a digital video coding and compression standard primarily used for web-based multimedia applications [6], utilises feature point data as body animation parameters to enhance object-based coding that ultimately facilitates data transmission and storage reduction.

Visual analysis of human motion in video images is a difficult problem in computer vision research. Though progress has been made in the last decade [7–9], marker-free video tracking is still in its infancy in many aspects [1]. Alternatively, marker-based optical motion capture (MoCap) systems are commercially available and have been widely used in the animation industry, such as the Vicon 512 [10]. In this case, motion and structure are presented solely by a small number of moving light displays (MLDs). Despite the complex imaging and vision processing for feature detection from images, we would argue high-level activity recognition information derived from the low-level feature data, such as the MLDs, can be coded more efficiently (than the raw MoCap files) as semantic indexing to enhance human-computer interaction and animation synthesis. Searching and browsing large MoCap file databases is difficult, if not impossible, unless each file is hand labelled with a descriptive string, for example, "run," "walk," and so forth. An interesting question, not only in the context of interaction analysis for computer

games, is how the categorisation and labelling of such data might be automated. If a solution capable of differentiating activities in real-time can be found, then there are also potential applications in interaction representation for games, with user movements controlling avatar animation. The accuracy of the body animation parameters at one extreme, and generic activity classes at the other, with network load of a remote server, for example, the deciding factor.

In this study, we concentrate on a high-level activity recognition task using hidden Markov models (HMMs). Therefore, our algorithm assumes the availability of feature point motion data that might be obtained by various methods and sensors, such as the 3D marker-based optical motion capture data used here. The rest of this paper is organised as follows. Section 2 reviews related work on activity recognition using HMMs. Section 3 describes our choice of feature vector and the use of HMMs for training and classification in the general case. Section 4 provides experimental results on the recognition of human activity. We discuss and conclude our work in Sections 5 and 6.

2. RELATED WORK

Bobick [11] describes three levels of motion understanding problem: *movement*, *activity*, and *action*. For the sake of clarity and cross comparison, we adopt the language of that framework here. The work presented addresses an *activity* recognition problem. We require knowledge of the various *movements* that form the *activities* and the temporal properties of the sequence. We do not attempt to address the questions of context and domain knowledge that allow for the description of *action*.

In the first application of HMMs to human motion recognition, Yamato et al. [12] classified a set of 6 different tennis strokes. They achieve good "familiar person" classification results (better than 90%) but recognition rates drop considerably when the test subject is removed from the training data. This work is also interesting for its use of hidden states with very short duration; they use 36 states for sequences that are between 23 and 70 symbols in length. Wilson and Bobick [13] adopt the HMM in their work on gesture recognition. They are able to recognise simple gestures such as a waving hand. They do not shape the topology of their state transition matrix, for example, by imposing a left-to-right structure on their trained HMM, but leave it potentially ergodic. They argue that although gestures may appear to us as a well defined sequence of conceptual states, they may appear to sensors as a complex mixture of perceptual states. This problem is addressed again by Campbell et al. [14] where the careful selection of features, for example, using velocity rather than position, results in a feature vector that approximates a prototypical trajectory through conceptual states when plotted out in feature space over time. They achieve good results classifying a variety of T'ai Chi moves, but all training and testing data is performed by the same individual, so the generality of the model is not evaluated. Bowden [15] shows that extracting a richer high dimensional feature vector and then performing dimensionality reduction with principal components analysis can help a model to generalise, alleviating the "familiar person" requirement. Brand and Hertzmann [16] introduce stylistic HMMs which specifically address this problem by attempting to recover the "essential structure" of data while disregarding its "accidental properties" in a separation of structure and style.

Brand [17] highlights shortcomings of HMMs for vision research, noting that many activities are not well described by the Markov condition, as they feature multiple interacting processes. He applies a coupled HMM to the classification of T'ai Chi movements, describing the interactions between both hands and shows improved performance over standard HMMs. Galata et al. [18] use variable length Markov models in order to dynamically vary the order of the Markov model. This allows for the consideration of shorter or longer state histories when analysing training data, facilitating the encoding of activity with correlations at different temporal scales.

Outside of the Markovian frameworks discussed in this section, other techniques have been successfully employed for human activity recognition. Section 4 of [1] gives a comprehensive review of the various techniques that have been applied to the action recognition task and a discussion of their relative merits. In particular, both template matching and neural networks have received much attention, for example, [19, 20], respectively. Template matching techniques offer low computational complexity and ease of implementation over state-space approaches such as the HMM. However, they are typically more sensitive to noise and variation in the speed of movements [1]. Neural networks have been shown to be an equally viable approach to human motion classification with near identical results to the HMM [21].

In the context of our own research, we are particularly interested in the HMM for its generative capabilities. The HMM is good for characterizing not only the spatial but also the temporal nature of data. Traversing a trained model gives believable synthetic data. In other work we use this feature of HMMs to provide predictions of a subject's movements in a markerless Bayesian tracking scheme. We believe that although the standard HMM undoubtedly entails consideration of the various shortcomings addressed by the approaches above, and others, it is still a powerful tool and has favourable training requirements versus some of its extensions.

3. METHOD

Human kinematic data used in this work was acquired using a Vicon 512, 3D marker-based optical motion capture system. This provides coordinates of markers attached to feature points on a subject, in the manner of a 3D-MLD system. Feature points are located on the head, torso, shoulders, elbows, wrists, hips, knees, and ankles. The data have been analysed before, with classification achieved by considering the data in the frequency domain [22].

3.1. Feature extraction

In a sequence of frames $m = 1, \ldots, M$ we select a subset of the available feature points. These were the markers on the right shoulder, elbows, wrists, right hip, knees and ankles. Angles

between right radius and right humerus, both radii, right femur and right tibia, and both tibia were then calculated.

For example, the angle between the two radii bones may be calculated from the marker location vectors $\mathbf{m}_{\text{Relb}}$, $\mathbf{m}_{\text{Rwri}}$, $\mathbf{m}_{\text{Lelb}}$, $\mathbf{m}_{\text{Lwri}}$ by defining limb vectors $\mathbf{l}_{\text{Lrad}} = \mathbf{m}_{\text{Lwri}} - \mathbf{m}_{\text{Lelb}}$ and $\mathbf{l}_{\text{Rrad}} = \mathbf{m}_{\text{Rwri}} - \mathbf{m}_{\text{Relb}}$. The relationship

$$|\mathbf{l}_{\text{Lrad}}||\mathbf{l}_{\text{Rrad}}|\cos\theta = \mathbf{l}_{\text{Lrad}} \cdot \mathbf{l}_{\text{Rrad}} \tag{1}$$

is then used to determine the angle θ between limbs. In this way, a feature vector is compiled at each frame (see Figure 1):

$$\mathbf{f}_m = \begin{pmatrix} \theta_{\text{Rrad,Lrad}} \\ \theta_{\text{Rhum,Rrad}} \\ \theta_{\text{Rfem,Rtib}} \\ \theta_{\text{Rtib,Ltib}} \end{pmatrix}, \qquad m = 1, \ldots, M. \tag{2}$$

As limbs are considered relative to one another, the feature vector should remain consistent for a particular pose regardless of the subject's location in the world coordinate system. Although the marker data is unavoidably noisy, this type of feature extraction will provide a tight coupling between conceptual and perceptual states.

3.2. Hidden Markov models

A hidden Markov model can be used to model a time series such as the one derived in the last section. This approach assumes that the underlying system is a Markov process, where the system's state at any timestep m is assumed to depend only on its state at $m - 1$. A standard Markov model is described by a set of states and a set of transition probabilities between these states. The state of the system is allowed to evolve stochastically and is directly observable. This approach may be extended with the introduction of a hidden layer between state and observer. Each state emits an observable symbol from an alphabet common to all states, according to some probability distribution over that alphabet (see Figure 2). This describes a system where both the evolution of the system and the measurement of that evolution are stochastic processes. In our own application HMMs are an appropriate tool as they allow us to handle both the natural variability in a human's performance of a particular activity and also the error of our sensors in estimating their movement.

In order to analyse experimental data using an HMM, we must train HMMs to represent a set of training data and then evaluate the probability that subsequent test data sets were produced by that model. In this way, we may classify a set of N distinct test activities using N HMMs. An HMM λ is specified by parameters S, A_{ij}, A_i, $\text{p}_i(\mathbf{f})$, where

(i) $S = \{s_1, \ldots, s_N\}$ is the set of hidden states;
(ii) the $N \times N$ matrix, A_{ij}, is the probability of a transition from state i to state j;
(iii) A_i is the probability of a sequence starting in state i;
(iv) $p_i(\mathbf{f})$ is the probability of observing feature vector $\mathbf{f}$ while in state i; the emission probability is modelled by a single multivariate Gaussian $p_i(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \boldsymbol{\mu}_i; \boldsymbol{\Sigma}_i) = \exp\{-1/2(\mathbf{f} - \boldsymbol{\mu}_i)^T / \sqrt{(2\pi)^D |\boldsymbol{\Sigma}_i|}\}$ with mean $\boldsymbol{\mu}_i$, covariance $\boldsymbol{\Sigma}_i$, and D the dimensionality of $\mathbf{f}$ (see Figure 3).

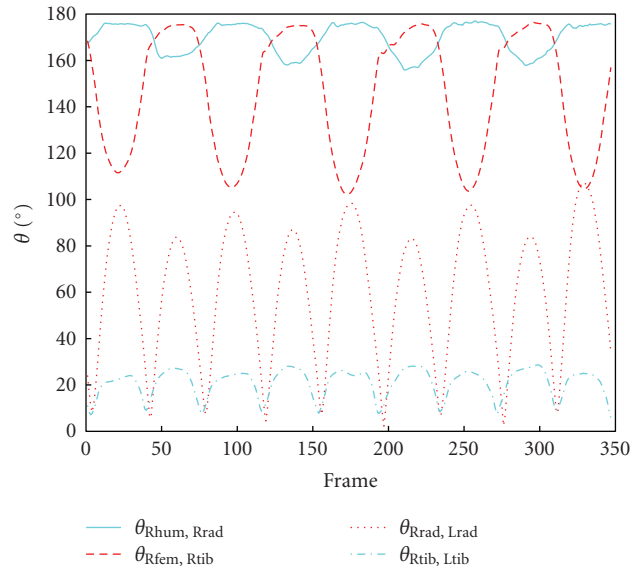

Figure 1: An example of the time series $\mathbf{f}$ for a walking subject.

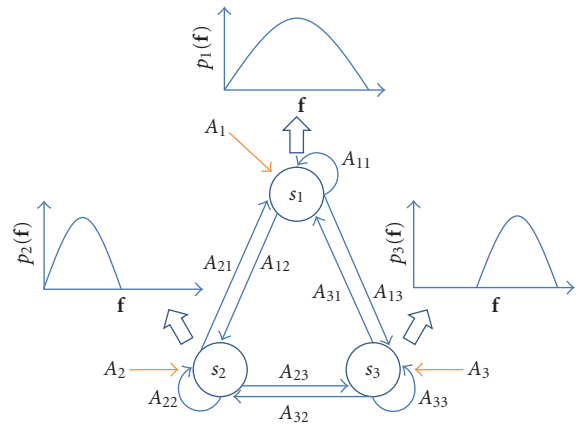

Figure 2: An example of a 3-state HMM with each state emitting a 1D feature vector $\mathbf{f}$.

Sections 3.3 and 3.4 give an overview of the use of continuous HMMs with single multivariate Gaussian observation functions for training and classification.

3.3. Training

Given a feature vector sequence $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_M\}$, we require the set of model parameters that maximise the probability that the data is observed. This problem cannot be solved analytically, but by making estimates of the initial model parameters and applying Baum-Welch reestimation, a form of expectation maximisation, iteration is guaranteed towards a local maximum in $p(F \mid \lambda)$ across the space of models. Although $p(F \mid \lambda)$ may contain a number of critical points, running the algorithm to convergence from a number of

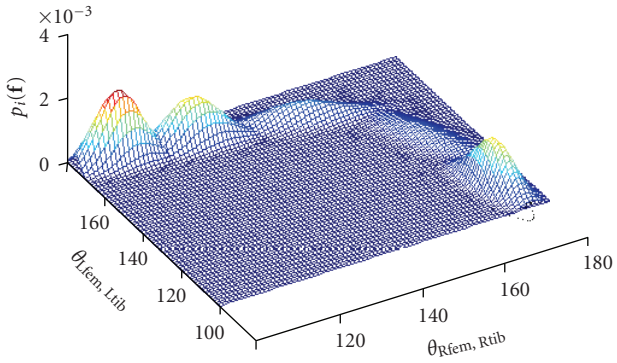

FIGURE 3: $\theta_{\text{Rfem,Rtib}}$ versus $\theta_{\text{Lfem,Ltib}}$ with 5 states.

different estimated initial conditions generally results in a good estimate of the global maximum [23].

The Baum-Welch algorithm requires calculation of the forward and backward variables for the data set F. The forward variable for a state i at time m is the total probability of all paths through the model that emit the training data up to time m, $\{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$ and finish in state i:

$$\alpha_{m,i} = p_i(\mathbf{f}_m) \sum_{j=1}^{N} \alpha_{m-1,j} A_{ji}, \tag{3}$$

where $\alpha_{1,i}$ is calculated using the distribution A_i, that is, $A_i p_i(\mathbf{f}_1)$. Similarly, the backward variable for a state i at time m is the total probability of all paths from state i that emit the rest of the training data $\{\mathbf{f}_{m+1}, \ldots, \mathbf{f}_M\}$:

$$\beta_{m,i} = \sum_{j=1}^{N} \beta_{m+1,j} p_j(\mathbf{f}_{m+1}) A_{ij}, \tag{4}$$

where $\beta_{M,i} = 1$. At any time m, the value $\alpha_{m,i}\beta_{m,i}$ gives the total probability of all paths through the model that produce the data F and pass through state i at time m. Furthermore, $\sum_{i=1}^{N} \alpha_{m,i}\beta_{m,i}$ is constant for all m and gives the probability of the sequence F given λ, or $p(F \mid \lambda)$. We can use these results to calculate the probability that the model was in state s_i when feature vector $\mathbf{f}_m$ was observed, given all the data:

$$\gamma_{m,i} = \frac{\alpha_{m,i}\beta_{m,i}}{\alpha_{m,i}\beta_{m,i}} \tag{5}$$

with which we can estimate the parameters of the Gaussian emission function $p(\mathbf{f})$ associated with each state i:

$$\begin{aligned} \boldsymbol{\mu}_i &= \frac{\sum_{m=1}^{M} \gamma_{m,i} \mathbf{f}_m}{\sum_{m=1}^{M} \gamma_{m,i}}, \\ \Sigma_i &= \frac{\sum_{m=1}^{M} \gamma_{m,i} (\mathbf{f}_m - \boldsymbol{\mu}_i)(\mathbf{f}_m - \boldsymbol{\mu}_i)^T}{\sum_{m=1}^{M} \gamma_{m,i}}, \end{aligned} \tag{6}$$

these are the first two maximisation steps.

In order to reestimate the matrix A_{ij}, we must consider the probability that a transition from state i to state j occurred between timesteps $m-1$ and m:

$$\xi_{m,ij} = p(q_m = s_j, q_{m-1} = s_i \mid F, \lambda) = \frac{\alpha_{m-1,i} A_{ij} p_j(\mathbf{f}_m) \beta_{m+1,j}}{p(F \mid \lambda)}, \tag{7}$$

where q_m is the active hidden state at time m. This is the total probability of all paths through the model which emit $\{\mathbf{f}_1, \ldots, \mathbf{f}_{m-1}\}$ and pass through state i at $m-1$ (given by α_{m-1}), multiplied by the transition-emission pair i transitions to j, j emits $\mathbf{f}_m$, multiplied by the total probability of all paths from state j that emit the remainder of the training data $\{\mathbf{f}_{m+1}, \ldots, \mathbf{f}_M\}$ (given by $\beta_{m,j}$), as a fraction of all paths through the model that emit the data.

By summing over the total number of state transitions, we get the expected number of transitions from i to j:

$$E_{ij} = \sum_{m=2}^{M} \xi_{m,ij}, \tag{8}$$

as the expectation step. The final maximisation step is then

$$A_{ij} = \frac{E_{ij}}{\sum_{j=1}^{N} E_{ij}}. \tag{9}$$

This process can then be iterated, with (6), and (9) providing the new estimate for λ, until some convergence criteria is met. A_i may also be reestimated as $\gamma_{1,i}$ although this is not done in this approach.

3.4. Classification

We can use the definition of the forward variable α in order to calculate the likelihood of a sequence of feature vectors given a particular set of model parameters. For a set of test data $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_M\}$ and model $\lambda = \{S, A_{ij}, A_i, p_i(\mathbf{g})\}$,

$$p(G \mid \lambda) = \sum_{i=1}^{N} \alpha_{M,i}. \tag{10}$$

Therefore, if an HMM is trained for each activity we are interested in recognising, we can evaluate the likelihood that unseen test data was emitted by each of the models and classify data as belonging to the model most likely to have produced it.

4. RESULTS

A set of 6 subjects were recorded performing 6 periodic activities using the Vicon system. These were walking on the spot, running on the spot, one-footed skipping, two-footed skipping, and two types of star jump. Each activity was performed by at least 3 individuals. Each sequence was divided into two halves, each of between 5 to 12 seconds at 60 fps. One half was used for training, the other retained for testing. Although the fact that the motions are periodic is useful as it negates the need to segment the training data, this is not a requirement of the approach. All of the steps described in Sections 4.1 and 4.2 were performed using the HMM Toolbox for Matlab [24].

Table 1: Classification of human activities.

	λ_{Jump1}	λ_{Jump2}	λ_{Run}	λ_{Skip1}	λ_{Skip2}	λ_{Walk}
G_{Jump1}	**17/20**	3/20	0/20	0/20	0/20	0/20
G_{Jump2}	0/20	**20/20**	0/20	0/20	0/20	0/20
G_{Run}	0/15	0/15	**15/15**	0/15	0/15	0/15
G_{Skip1}	0/15	1/15	0/15	**12/15**	2/15	0/15
G_{Skip2}	0/20	0/20	0/20	0/20	**20/20**	0/20
G_{Walk}	0/15	0/15	0/15	0/15	0/15	**15/15**

4.1. Activity training

A feature vector was extracted at each frame as described in Section 3.1. This vector was then extended to contain a finite difference estimate of $\Delta\mathbf{f}_m$ made using the previous timestep, that is, $\Delta\mathbf{f}_m \approx \mathbf{f}_m - \mathbf{f}_{m-1}$. This is helpful in resolving ambiguities such as intersections in the feature vector trajectory, thus reducing the number of states that represent a junction in feature space. It is analogous to a second order HMM, where the previous state as well as current state have an effect on the next transition, thus encapsulating extra "history" in each state of a first order HMM. Each of the activities was represented by 30 states. As in [12] this is a relatively large number considering that each activity has a period of approximately one second. Emitting consecutive conceptual state vectors from the mean point of each state will produce almost identical poses. However, a large number of states helps the initial clustering and provides good results even if it is not intuitively appealing [25].

Initial estimates of the state means and covariance matrices were found by K-means clustering [26]. The transition matrix was initially estimated randomly (with each row of A_{ij} summing to 1) and the prior A_i set with every value equal to $1/N$, where N is the total number of states. A_i was not reestimated in order that test data could begin at any point during the activity unit with no probabilistic penalty. The transition probabilities and state means and covariances were reestimated using no more than 20 iterations of the Baum-Welch update equations of Section 3.3.

4.2. Activity classification

Each subject's test data for each activity was tested separately. Feature vectors were again extracted at each frame to build up a set of observations G. $p(G \mid \lambda)$ was then calculated 5 times for each test sequence, the Baum-Welch algorithm having been allowed to reconverge to a newly estimated set of parameters λ each time. Table 1 summarises the classification results for each batch of activity test data against each trained model. For cross comparison, the forward variable is calculated over the first 2.5 seconds of each test sequence ($M = 150$ in (10)). Classification results are concentrated on the diagonal and no misclassifications are made for 4 of the activities. In the cases of Jump1 and Skip1, all off-diagonal classifications are due to just one test sequence in each batch, with all other sequences being correctly classified. Further discussion of these results is given in Section 5. Using the HMM Toolbox for Matlab, evaluation of $p(G \mid \lambda)$ typically takes between 0.05 to 0.08 seconds, facilitating real-time calculation of $p(G \mid \lambda)$ for the 6 HMMs.

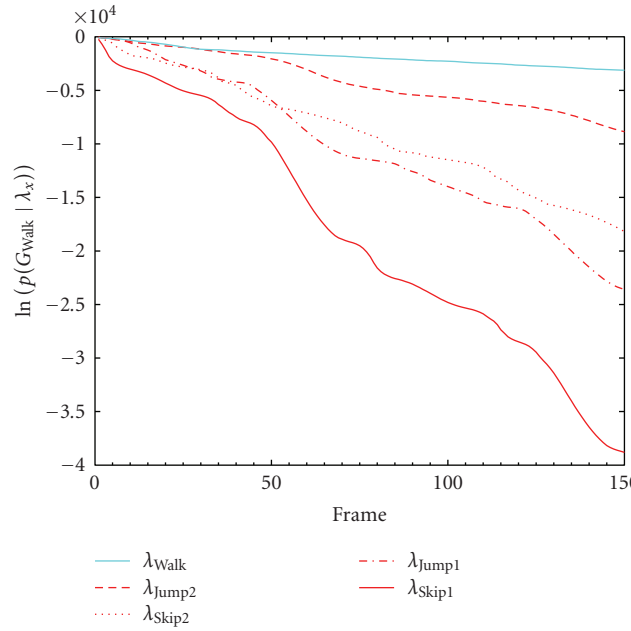

Figure 4: Forward variable for one subject's test walking sequence for all activity models as a function of the number of frames (m).

4.3. Confusion matrices

In the framework outlined in [2], a key aspect of any gesture based interface is its speed in determining a user's activity. Although $p(G \mid \lambda)$ may be calculated in real-time, any approach is limited by the need for sufficient data to stabilise the results of the forward variable evaluations. Determining this data requirement is key to quantifying the level of latency introduced to game play by a gesture based interface. Figure 4 shows the forward variable evaluated using one subject's test walking sequence for each activity model as a function of the number of frames taken as input (m). $p(G_{\text{Walk}} \mid \lambda_{\text{Run}})$ caused arithmetic overflow at $m = 2$ and is not plotted. Walking is not correctly established as the most likely activity until $m = 4$ and jumping temporarily overtakes it for $m = 27, 28, 29$. Walking subsequently remains the most likely interpretation.

Table 2: Classification of two-footed skipping activity versus data segment length.

	λ_{Jump1}	λ_{Jump2}	λ_{Run}	λ_{Skip1}	λ_{Skip2}	λ_{Walk}
$M = 2$	0.0114	0.0193	0.0000	0.0386	**0.9277**	0.0000
$M = 4$	0.0089	0.0114	0.0000	0.0309	**0.9495**	0.0000
$M = 8$	0.0017	0.0017	0.0000	0.0017	**0.9950**	0.0000
$M = 16$	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000
$M = 32$	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000
$M = 64$	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000

In order to determine how quickly reliable classification may take place across the activity cycle, each training sequence was divided into smaller segments for evaluation with the forward variable. Segment lengths of 2, 4, 8, 16, 32, and 64 frames were used and all possible continuous segments of this length tested, with data segments allowed to overlap, thus maximising the number of classification problems considered. The classification results are used to form a confusion matrix for each activity. The confusion matrix for the two-footed skipping activity is shown in Table 2. A correct classification rate of greater than 99% is achieved with a segment size of 8 frames, equivalent to 0.13 seconds of data.

5. DISCUSSION

The learned transition matrices A_{ij} were strongly focused on just a few columns per row. As in [15], no effort was made to number the states meaningfully, for example, in chronological order. However, it would suggest that even though no topology shaping was attempted, Baum-Welch training found a natural left-to-right type structure for the HMM where each state may be self-referential, or may transition to a handful of nearby (in terms of the feature space) states. This supports the claim that the feature vector achieves a tight level of coupling between the conceptual and perceptual states.

Classification between the broad activity types (run, walk, skip, jump) was reliable, but subtle changes in the activity proved harder to classify. For example, the confusion between the two star jumps and one-footed and two-footed skipping seen in the first and fourth row of Table 1 respectively. These activities were only misclassified for one individual's test sequence in each case, and in the case of skipping we believe this to be due to a lack of training data for that subject, causing Baum-Welch training to overfit to the other, longer sequences. However, in the case of the star jumping, the similarity between the two activities, in terms of the feature vector we extract, may mean they are unsuitable for inclusion in a gesture interface as a pair. Included separately, they do not pose a problem.

The compilation of confusion matrices demonstrated that classification was feasible with the consideration of only small amounts of data. The reduction of segment length produced remarkably little spread in the distribution across activity columns of the matrix. Balancing the tradeoff between accuracy and latency in a gesture based interface is an application dependent decision, but confusion matrices compiled in this way should facilitate such development decisions.

Although the models performed well when the individual concerned formed part of the training group, performance worsened significantly when they were removed. Only running on the spot and walking on the spot were consistently recognised. This drop in performance is broadly in line with previous findings, for example, [12]. The resulting models may have failed to recover "underlying structure" due to the high level of variation between training data. Alternatively, they may have suffered from overfitting to what is a small set of training data and an impoverished representation of the activity. In either case, a larger number of people in the training set should improve results.

6. CONCLUSIONS

We have described a technique for classifying human activities with HMMs. In this baseline study, buffered marker data obtained from a MoCap system were successfully used for human activity analysis in real-time. These results demonstrate the proposed method remains a candidate for feature-based on-line recognition tasks in gesture based games.

Although MoCap data is used here, the doubly stochastic nature of the HMM should allow for the use of less invasive, but more noisy, markerless tracking techniques. The HMM may provide a way of interpreting complex user input available from a new generation of computer game input devices, providing a more natural and engaging user experience. This type of high level semantic description of a person's movements could also be incorporated into object based coding schemes such as body animation parameters, as an activity index for decoders.

ACKNOWLEDGMENTS

This research was made possible by an MMU Dalton Research Institute research studentship and EPSRC Grant EP/D054818/1. All MoCap data used in this paper were obtained by an optical motion capture system installed at the Department of Computer Science, University of Wales, UK.

REFERENCES

[1] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[2] W. T. Freeman, D. B. Anderson, P. A. Beardsley, et al., "Computer vision for interactive computer graphics," *IEEE Journal of Computer Graphics and Applications*, vol. 18, no. 3, pp. 42–53, 1998.

[3] A. Menache, *Understanding Motion Capture for Computer Animation and Video Games*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1999.

[4] J. Starck, G. Miller, and A. Hilton, "Video-based character animation," in *Proceedings of the ACM SIGGRAPH Eurographics Symposium on Computer Animation (SCA '05)*, pp. 49–58, Los Angeles, Calif, USA, July 2005.

[5] J. Wilhelms and A. V. Gelder, "Interactive video-based motion capture for character animation," in *Proceedings of the IASTED Conference on Computer Graphics and Imaging (CGIM '02)*, Kauai, Hawaii, USA, August 2002.

[6] Joint Video Team, Information technology—coding of audio-visual objects—part 10: advanced video coding, MPEG-4, ISO/IEC 14496-10, 2005.

[7] A. O. Bălan, L. Sigal, and M. J. Black, "A quantitative evaluation of video-based 3D person tracking," in *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS '05)*, vol. 2005, pp. 349–356, Beijing, China, October 2005.

[8] N. Jojic, M. Turk, and T. S. Huang, "Tracking self-occluding articulated objects in dense disparity maps," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 123–130, Kerkyra, Greece, September 1999.

[9] C. Sminchisescu and B. Triggs, "Kinematic jump processes for monocular 3D human tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 69–76, Madison, Wis, USA, June 2003.

[10] Vicon motion systems, http://www.vicon.com/.

[11] A. Bobick, "Movement, activity and action: the role of knowledge in the perception of motion," in *Royal Society Workshop on Knowledge-Based Vision in Man and Machine*, pp. 1257–1265, London, UK, February 1997.

[12] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '92)*, pp. 379–385, Champaign, Ill, USA, June 1992.

[13] A. D. Wilson and A. F. Bobick, "Learning visual behavior for gesture analysis," in *Proceedings of International Symposium on Computer Vision (ISCV '95)*, pp. 229–234, Coral Gables, Fla, USA, November 1995.

[14] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick, and A. Pentland, "Invariant features for 3-D gesture recognition," in *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG '96)*, pp. 157–162, Killington, Vt, USA, October 1996.

[15] R. Bowden, "Learning statistical models of human motion," in *Proceedings of the IEEE Workshop on Human Modeling, Analysis and Synthesis (CVPR '00)*, pp. 10–17, Hilton Head Island, SC, USA, July 2000.

[16] M. Brand and A. Hertzmann, "Style machines," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 183–192, New Orleans, La, USA, July 2000.

[17] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 994–999, San Juan, Puerto Rico, USA, June 1996.

[18] A. Galata, N. Johnson, and D. Hogg, "Learning variable-length Markov models of behavior," *International Journal of Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 398–413, 2001.

[19] A. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 39–42, Sarasota, Fla, USA, December 1996.

[20] Y. Guoa, G. Xu, and S. Tsuji, "Understanding human motion patterns," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR '94)*, vol. 2, pp. 325–329, Jerusalem, Israel, October 1994.

[21] I. Boesnach, J. Moldenhauer, C. Burgmer, T. Beth, V. Wank, and K. Bos, "Classification of phases in human motions by neural networks and hidden markov models," in *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems (CCIS '04)*, vol. 2, pp. 976–981, Singapore, December 2004.

[22] B. Li and H. Holstein, "Recognition of human periodic motion-a frequency domain approach," in *Proceedings of the International Conference on Pattern Recognition (ICPR '02)*, vol. 1, pp. 311–314, Quebec, Canada, August 2002.

[23] A. B. Poritz, "Hidden Markov models: a guided tour," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '88)*, pp. 7–13, New York, NY, USA, April 1988.

[24] K. Murphy, Hidden Markov model toolbox for Matlab http://www.cs.ubc.ca/~murphyk/software/HMM/hmm.html.

[25] D. O. Tanguay Jr., "Hidden Markov models for gesture recognition," M.S. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1995.

[26] A. Gersho, "On the structure of vector quantizers," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 157–166, 1982.