Modelling 'Talking Head' Behaviour

Craig Hack and Chris Taylor
Imaging Science
The University of Manchester
Manchester, M13 9PT, UK
craig.hack@stud.man.ac.uk

Abstract

We describe a generative model of 'talking head' facial behaviour, intended for use in both video synthesis and model-based interpretation. The model is learnt, without supervision, from talking head video, parameterised by tracking with an Active Appearance Model (AAM). We present a integrated probabilistic framework for capturing both the short-term visual dynamics and longer-term behavioural structure. We demonstrate that the approach leads to a compact model, capable of generating realistic and relatively subtle talking head behaviour in real time. The results of a forced-choice psychophysical experiment show that the quality of the generated sequences is significantly better than that obtained using alternative approaches, and is indistinguishable from that of the original training sequence.

1 Introduction

This paper addresses the problem of modelling the facial behaviour of individuals engaged in conversation. The aim is to develop a generative model capable of synthesising realistic, completely novel video sequences of conversational behaviour. The resulting model is of immediate application in video synthesis, but our long-term interest is in model-based interpretation and human-computer interaction – extending into the temporal domain the 'interpretation by synthesis' strategy that has been applied successfully to static face images [7, 3].

In order to capture a broad range of often subtle behaviours, we choose to learn a model from very long (around one hour) video sequences of individuals in conversation. To be practical, this requires that we adopt an unsupervised (or very nearly unsupervised) approach. Our work builds on the Active Appearance Model (AAM) method of Cootes *et al*. [3], and on several previously published techniques for modelling behaviour. We present a new approach to combining short-term modelling of appearance dynamics, with longer-term modelling of behavioural structure, achieving realistic synthesis of subtle and highly variable facial behaviour.

2 Previous Work

There is a significant body of literature on modelling visual behaviour. Though relatively little of this applies specifically to facial behaviour, the methodology is relevant. Common

BMVC 2003 doi:10.5244/C.17.4

to all approaches is the idea of generating a sequence of states through some configuration space.

Building on experience from speech recognition, Hidden Markov Models (HMMs) have been applied extensively. Yamoto *et al*. [13] were amongst the first to propose the approach. Starner and Pentland [12] used Gaussian HMMs to recognise American Sign Language, and many others have followed. Although such methods show promise for the recognition of simple gestures, the HMMs they use are not very satisfactory generative models: they make poor predictions because they use a very limited state history and draw from each state distribution independently.

Brand [1] addressed the state sampling issue by modelling both configuration variables and their derivatives, allowing the estimation of a maximum likelihood path through configuration space. As with conventional HMM methods, this approach does not, however, use sufficient history to capture configuration dynamics. Johnson and Hogg [6] extended this approach (building also on work by Jebara and Pentland [5]), sampling each step along the trajectory in configuration space from a distribution conditioned on its recent history. They showed quite good results in a simple problem of modelling the paths of pedestrians in fixed environment, but the approach does not capture the full structure of trajectories in behaviour space, and produces relatively poor results in our application.

Other authors have described auto-regressive models, that aim, specifically, to capture short-term dynamics. Fitzgibbon [4] showed that temporally coherent behaviour could be generated using an ARMA model. Campbell *et al*. [2] extended the autoregressive approach to deal with non-stationary behaviour, and applied the approach (with limited success) to a talking head. The problem with these methods is that, although they reproduce appropriate short-term dynamics, they do not capture longer-term behavioural structure.

More recently, exemplar-based approaches have become popular. Schödl *et al*. [10] find a set of prototype (key) frames from an image sequence and construct new sequences by transitioning between similar prototypes. Sidenbladh *et al*. [11] extend this idea, by creating prototypes that are short sequences of frames. To select the next frame, a set of prototypes are found whose histories are similar to the current history; one of these is selected, and its final frame is used as the next frame. These approaches produce quite convincing results, but do not have the ability to generalise significantly.

Our approach is related to that of [5] and [6], but we capture more of the structure in behaviour space by using an HMM. This requires the sampling approach to be extended to condition state transitions, as well as the sampled behaviour, on the current value of the behaviour vector.

3 Overview of Our Approach

Our aim is to model the distribution of facial behaviours seen in a long training sequence, capturing both short-term dynamics and longer-term behavioural structure. We use an Active Appearance Model (AAM) to parameterise talking-head image sequences, and build behaviour models in the space of this parameterisation. Novel video sequences are reconstructed from a stream of Appearance Model parameters generated by the behaviour model.

We capture short-term dynamics by modelling sequences of parameter vectors, which

we call pathlets. Each pathlet is a point in a high dimensional space. We model the behavioural structure of this space using a Gaussian HMM. It is clear that successive pathlets cannot be chosen independently – otherwise temporal coherence would be lost at the joins. We show how the choice of each pathlet can be conditioned on its predecessor, delivering consistent short-term dynamics, and properly integrating short-term and longer-term behaviour.

4 AAM Parameterisation

To reduce the dimensionality of the learning problem, we model behaviour in the *Appearance Space* of an Active Appearance Model (AAM). An Appearance Model represents both the shape and texture variability seen in a training set. The training set consists of images, with corresponding landmark points marked on each example. The configuration of land-mark points for each example can be represented by a vector $\mathbf{x}$, and the texture – warped to the mean shape and raster-sampled – by a vector $\mathbf{g}$. The appearance model has a vector of *appearance parameters*, $\mathbf{c}$, that controls the shape and texture according to:

$$\begin{aligned} \mathbf{x} &= \overline{\mathbf{x}} + \mathbf{Q}_s \mathbf{c}. \\ \mathbf{g} &= \overline{\mathbf{g}} + \mathbf{Q}_g \mathbf{c}. \end{aligned} \tag{1}$$

where $\overline{\mathbf{x}}$ is the mean shape, $\overline{\mathbf{g}}$ is the mean texture and $\mathbf{Q}_s$,$\mathbf{Q}_g$ are matrices, derived from the training set, describing modes of shape and texture variation respectively. A new face image can be synthesised for a given parameter vector, $\mathbf{c}$, by generating a texture image from the vector $\mathbf{g}$ and warping it using the control points described by $\mathbf{x}$. Typically, good quality face images with a wide range of poses and expressions can be synthesised using a model with an appearance vector $\mathbf{c}$ containing around 100 appearance parameters.

An appearance model can be matched to a new image, given an initial approximation to the position, using the Active Appearance Model (AAM) algorithm. This uses a fast linear update scheme to modify the model parameters, minimising the difference between the synthesized image and target image. If an AAM is matched to each image in a video sequence of a talking head, the facial behaviour can be described by a sequence of $\mathbf{c}$ vectors, one for each frame.

5 Modelling Behaviour

5.1 Pathlets and Pathlet Space

To capture short-term behaviour, we construct pathlets from sequences of appearance parameters. The simplest scheme would be to take the values of $\mathbf{c}$ for successive frames, but that would result in many redundant pathlets, where there was no change in the appearance parameters between frames. Instead we choose to sample at points spaced equally in appearance space. In the experiments described below, we fitted a cubic spline to the sequence of $\mathbf{c}$ values and sampled at points equally spaced along this continuous trajectory, such that the average time between samples was equal to the interval between frames in the original sequence. Each of these points was parameterised by

$$\mathbf{x}_i = [\mathbf{c}_i, at_i] \tag{2}$$

where t_i is the logarithm of the time taken to travel from $\mathbf{c}_i$ to $\mathbf{c}_{i+1}$ and a is a constant, chosen to make the variance of $\mathbf{c}_i$ and t_i comparable over a long sequence. A complete video sequence $\mathbf{X}$ can be represented by a series of non-overlapping pathlets of length l:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{points}}]^T = [\lambda_1^T, \lambda_2^T, \ldots, \lambda_{N_{paths}}^T]^T \tag{3}$$

where $\lambda_n = [\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+l}]^T$, and $n = 1 + (i-1)/l$.

It is useful to think of pathlets as shapes in Appearance space. Each of these shapes can be represented as a point in an $dim(\mathbf{x})l$ dimensional space, where $dim(\mathbf{x})$ is the dimensionality of $\mathbf{x}$. It is clear that there will tend to be strong correlations between the values of $\mathbf{x}_i$ within a pathlet. To exploit this and reduce the dimensionality of the space, we perform Principal Component Analysis (PCA) on the set of pathlets from a training sequence. It is in this PCA space that we build our behaviour model; to simplify notation, we shall assume, from this point on, that λ_i is a vector in PCA space and we will refer to the PCA space as *pathlet space*. For reasons that we shall see shortly, we also define *long-pathlets*, obtained by concatenating successive pathlets:

$$\Lambda_{n+1} = [\lambda_n^T, \lambda_{n+l}^T]^T \tag{4}$$

5.2 An HMM in Pathlet Space

Pathlets provide a representation capable of capturing the short-term dynamics of visual behaviour. We now consider the problem of capturing the longer-term structure of behaviour. Our approach is based on constructing a Hidden Markov Model (HMM) in pathlet space. In this section we outline the standard approach to constructing an HMM and using it to generate novel sequences; we also explain the limitations of this method. In the following section we present a modified approach that overcomes these problems. We use a Gaussian HMM, which can be thought of simply as a Gaussian mixture model with additional sequencing information. We provide here a brief summary of HMM construction and use. For a more detailed description see Rabiner [9].

Our starting point in constructing an HMM is a sequence of pathlets $\lambda_1, \lambda_2, \ldots, \lambda_{N_{paths}}$ obtained from a long training sequence of talking head video. The Baum-Welch algorithm (a specialised case of the EM algorithm) can be used to obtain a HMM that is characterised by:

1. A set of states $\{s_k\}$. We refer to the n^{th} state in a sequence of states as s_{q_n}.
2. A Gaussian distribution $N[\lambda|\mu_k, \mathbf{K}_k]$ in pathlet space for each state s_k, with mean μ_k and covariance $\mathbf{K}_k$.
3. A k x k state transition matrix $\mathbf{A} = \{a_{j,k}\}$, where $a_{j,k} = P(s_k|s_j)$. Typically $\mathbf{A}$ is not ergodic, hence many $a_{j,k} = 0$.
4. The initial state distribution $\Pi = \{\pi_j\}$, where $\pi_j = P(q_1 = s_j)$.

Such an HMM can be used as a stochastic generative model, creating behaviours that are consistent with those seen during training. Following the standard approach we proceed as follows:

1. Choose an initial state $s_{q_n}, n = 1$ by sampling from the set of states $\{s_k\}$ with probabilities π_k.
2. Sample a pathlet λ_n from $N[\lambda|\mu_{q_n}, \mathbf{K}_{q_n}]$.
3. Choose a transition to a new state q_{n+1} by sampling from the set of states $\{s_k\}$ with probabilities $a_{q_n,k}$.
4. Repeat from 2.

Each sample of λ_n in step 2 provides l time points in appearance space, which can be concatenated to produce an evolving trajectory. The mixture of Gaussians, $\{N[\lambda|\mu_{q_n}, \mathbf{K}_{q_n}]\}$, in pathlet space captures the distribution of legal short-term behaviour, whilst the constrained transitions between states capture the longer-term structure of behaviour.

This scheme does not, however, lead to satisfactory results in our application (see experimental results in section 6.3). There are two closely related problems: first, in step 2, we should not pick a value for λ_n independently of the previous pathlet, because this does not guarantee that successive pathlets will join continuously; second, in step 3, we should not pick state q_{n+1} independently of λ_n, because the probability of a transition to a given state depends not only on the current state, but also the current pathlet. The first point is reasonably obvious. The second point is illustrated in figure 1(a), which shows the distribution of training pathlets for one state, plotted in the first three dimensions of pathlet space. Each point is labelled to show its destination state; there is a clear correlation between position within the distribution and destination state.

5.3 A Joint Pathlet HMM

In this section we introduce a new method – the Joint Pathlet HMM (JPHMM) – that deals with the problems we identified above with the simple pathlet HMM scheme outlined.

In order to capture the conditional relationships between successive pathlets, we start by building an HMM in long pathlet space, rather than pathlet space. The method is exactly as described above, except that the starting point is a sequence of long pathlets $\Lambda_2, \Lambda_3, \ldots, \Lambda_{N_{paths}}$ (Λ_1 is not defined). Otherwise all notation remains the same. Note that these long pathlets overlap by one standard pathlet, so each step introduces just one new pathlet.

To generate legal behaviour we need to sample from $p(\lambda_{n+1}|\lambda_n, q_n)$, the conditional distribution of λ_{n+1}, given the current state s_{q_n} and current pathlet λ_n. This conditional distribution can be expressed as a weighted sum of the corresponding conditional distributions associated with each of the states:

$$p(\lambda_{n+1}|\lambda_n, q_n) = \sum_k P(q_{n+1} = k|\lambda_n, q_n) p(\lambda_{n+1}|\lambda_n, q_{n+1} = k) \tag{5}$$

The procedure for sampling from this distribution is to select a state $s_{q_{n+1}}$ from all possible states $\{s_k\}$ with probabilities $P(q_{n+1} = k|\lambda_n, q_n)$, then to sample from $p(\lambda_{n+1}|\lambda_n, q_{n+1} = k)$.

Using Bayes Theorem, the first of these terms can be expanded as follows:

$$P(q_{n+1} = k|\lambda_n, q_n) = \frac{P(q_{n+1} = k|q_n) p(\lambda_n|q_{n+1} = k, q_n)}{p(\lambda_n|q_n)} \tag{6}$$

Since the denominator is independent of q_{n+1}, the selection between the different possibilities for q_{n+1} can be made by weighting the decision using the numerator alone. The first term in the numerator is simply the appropriate entry in the transition matrix **A**, obtained during construction of the HMM. The second term can be learnt from the training set, once the HMM has been constructed. To do this, we use the HMM to find the most probable sequence of states giving rise to the observed data, using the Viterbi algorithm [9]. For each state transition occurring with non-zero probability, we find a Gaussian approximation to $p(\lambda_n|q_{n+1},q_n)$ from the training data.

Once we have chosen state q_{n+1} using equation 6, we need to sample from $p(\lambda_{n+1}|\lambda_n,q_{n+1})$, the distribution of pathlets for state q_{n+1}, conditioned on λ_n. This is illustrated in Figure 1(b). To sample from $p(\lambda_{n+1}|\lambda_n,q_{n+1})$ we use a similar sampling scheme to that of Johnson and Hogg [6]. From our HMM training we have

$$p(\Lambda_{n+1}|q_{n+1}) = N[\Lambda|\mu_{q_{n+1}},\mathbf{K}_{q_{n+1}}] \tag{7}$$

where $\Lambda_{n+1} = [\lambda_n^T, \lambda_{n+l}^T]^T$. We can decompose $\mu_{q_{n+1}}$ and $\mathbf{K}_{q_{n+1}}$ into the components that correspond to λ_n and λ_{n+1}:

$$\mu_{q_{n+1}} = \begin{bmatrix}\mathbf{m}_{q_n} & \mathbf{m}_{q_{n+1}}\end{bmatrix}. \tag{8}$$

$$\mathbf{K}_{q_{n+1}} = \begin{bmatrix}\mathbf{k}_{q_n,q_n} & \mathbf{k}_{q_{n+1},q_n}^T \\ \mathbf{k}_{q_{n+1},q_n} & \mathbf{k}_{q_{n+1},q_{n+1}}\end{bmatrix}. \tag{9}$$

The conditional density we require is also Gaussian, $N[\lambda|\mu_{q_{n+1}|q_n},\mathbf{K}_{q_{n+1}|q_n}]$ [8], where:

$$\mu_{q_{n+1}|q_n} = \mathbf{m}_{q_n} + \mathbf{k}_{q_{n+1},q_n}\mathbf{k}_{q_n,q_n}^{-1}(\lambda_{q_n} - \mathbf{m}_{q_n}). \tag{10}$$

$$\mathbf{K}_{q_{n+1}|q_n} = \mathbf{k}_{q_{n+1},q_{n+1}} - \mathbf{k}_{q_{n+1},q_n}\mathbf{k}_{q_n,q_n}^{-1}\mathbf{k}_{q_{n+1},q_{n+1}}^T. \tag{11}$$

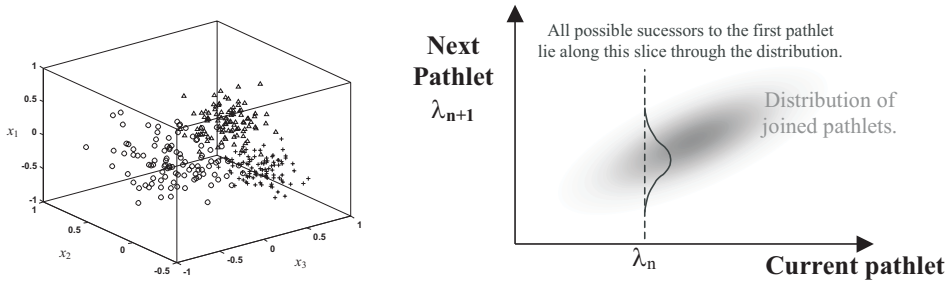
In summary, we can generate novel behaviour sequences using the JPHMM as follows:

1. Choose an initial state s_{q_n}, $n = 1$ by sampling from the set of states $\{s_k\}$ with probabilities π_k. Sample a long pathlet Λ_n from $N[\Lambda|\mu_{q_1},\mathbf{K}_{q_1}]$.
2. Choose a transition to a new state q_{n+1} by sampling from the set of states $\{s_k\}$ with probabilities $P(q_{n+1} = k|\lambda_n,q_n)$ given by Equation 6.
3. Sample from $p(\lambda_{n+1}|\lambda_n,q_{n+1})$ using equations 7, 8, 9, 10 and 11.
4. Repeat from 2.

6 Experiments

6.1 Training Data

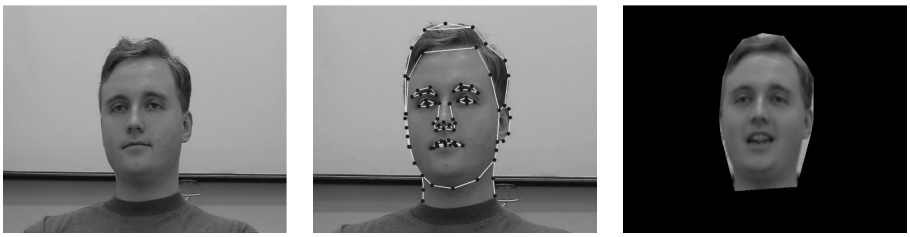
We recorded a video sequence, lasting approximately 1 hour, of one of two people involved in a conversation. We then used a bootstrapping method to achieve reliable tracking of the whole sequence. Several frames from the sequence were marked up initially with corresponding points and used to train an Active Appearance Model (AAM). The



(a) (b)

Figure 1: (a) Scatter plot of training data, in pathlet space, for one state. Points are labelled according to destination state using symbols o, +, $\bigtriangledown$. (b) Finding the PDF of pathlet λ_{n+1} given a previous pathlet in the sequence.

trained AAM was then used to match to each frame in the sequence, using the result from the previous frame to initialise the search. The mean square matching errors were then analysed to determine on which frames the AAM failed. The 10 worst frames were marked up and included in the set used to train the AAM. The newly trained AAM was again used to track the face in the video sequence. This process was iterated until the worst frames achieved acceptable tracking performance. Figure 2(a) shows a typical image from the recorded sequence. Figure 2(b) shows the points used to mark up the images to train the AAM.



(a) (b) (c)

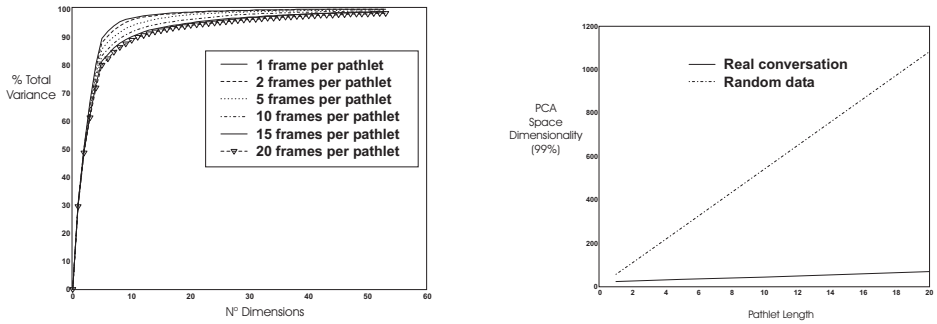
Figure 2: (a) A sample image taken directly from the video used to train the model. (b) An image showing the points used to train the AAM. (c) A reconstruction of a marked up image from its parameter values.

6.2 Model Building

88,000 frames of the conversation video were tracked and parameterised using the bootstrapped AAM as described above. A 53 dimensional AAM space was used to model the variation in the video sequence, retaining 98% of the total model variance. Figure 2(c)

shows a frame reconstructed from its parameter values. Pathlets were extracted as described in Section 5.1. To make maximum use of the training data, l sequences of pathlets were used, starting from each of the first l points in the training sequence.

To investigate the effect of changing pathlet length, we compared the distribution of variances in the PCA components of the pathlet space. Figure 3(a) shows the PCA variance spectra for pathlet lengths of 1,2,5,10,15 and 20. As would be expected for a structured sequence, the first few dimensions of the pathlet space capture most of the pathlet variance.



(a) (b)

Figure 3: (a) Total variance retained as a function of number of PCA dimensions for different lengths of pathlet; (b) number of dimensions required to retain 99% of the total variance as a function of pathlet length.

Figure 3(b) shows the number of dimensions in pathlet PCA space required to capture 99% of the total pathlet variance, as a function of pathlet length. To highlight the effectiveness of using pathlets to encode short-term dependencies we have included the equivalent results for pathlets constructed from random data. The difference in the gradients of the two lines shows clearly that pathlets capture useful local structure.

Based on these results, we chose pathlets of length 10 (typically half a second) as a good compromise - capturing useful dynamics whilst avoiding creating a space with so many dimensions that training becomes impractical for the size of training set available. We retained 99% of the total pathlet variance, producing a pathlet PCA space with only 30 dimensions. An HMM with 80 hidden states was trained in the long-pathlet space as described above.

6.3 Model Performance

We used the model as described in Section 5 to generate new sequences of talking head behaviour, and found that the results were reasonably convincing. To assess the performance quantitatively, we performed a forced-choice psychophysical experiment, in which subjects were presented with pairs of video sequences of conversational behaviour, and were asked to choose the most realistic. We used this approach to compare sequences

Comparison	Total Comparisons	% JPHMM	χ^2	Significance
Original Sequences	218	54	1.33	0.75
Appearance HMM	188	99	178.13	1.00
Pathlet HMM	209	91	141.55	1.00
Smoothed Pathlet HMM	205	84	92.90	1.00

Table 1: Results of psychophysical experiment comparing the ability of subjects to distinguish between the realism of JPHMM results, the original training sequence, and the output of 3 other modelling methods.

generated using our JPHMM method with examples of the original training data and with sequences generated using three simpler modelling methods.

The forced choice experiment was performed by 20 subjects, each making 40 comparisons. For each comparison both sequences were played simultaneously for 10 seconds. The subject then had 5 seconds to choose the most realistic sequence using a point and click interface. Each subject was given written instructions and each session was structured as follows: 3 practice comparisons, 2 minute break for questions, first batch of 20 recorded comparisons, 2 minute rest, second batch of 20 recorded comparisons. A pool of 46 examples of each type of sequence was used, and pairs suitable for one of the four types of comparison considered were selected randomly, with equal probability, for each subject. The total number of comparisons was 800 giving approximately 200 trials for each type of comparison. The four types of comparison used were: JPHMM vs Original Sequence, JPHMM vs Appearance HMM, JPHMM vs Pathlet HMM and JPHMM vs Smoothed Pathlet HMM.

The Original Sequences were video sequences, resynthesised from the parameterised training data and thus represented the most realistic possible results. The Appearance HMM was an HMM with 300 states, built directly using the parameterised training sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{points}}]^T$. Since this involved no history, there was no mechanism for capturing short-term behavioural dynamics, so relatively poor performance was expected. The simple Pathlet HMM was built as described in section 5.2, using 200 states. Although the short-term dynamics were expected to be captured, the problems with pathlet continuity, outlined earlier, were expected to lead to overall poor performance. The Smoothed Pathlet HMM was identical to the previous case, except that a spline was constructed in appearance space and the sequence was synthesised by sampling points along this smooth trajectory. This was included to test whether a relatively simple modification of the simple pathlet method could give acceptable results.

The results of the experiment are summarised in Table 1. A chi-squared test was used to test the hypothesis that subjects could not distinguish between the pairs of sequences used in each type of comparison. The results show that our new method performs extremely well, and is much better than any of the alternatives investigated. Subjects were unable to distinguish between the JPHMM sequences and the Original Sequences (results favour the JPHMM, but are only significant at the 75% level). For all other methods, Subjects found the JPHMM sequences more realistic than those generated using all the other methods, with a very high level of statistical significance. Of the other methods, the Smoothed Pathlet HMM method performed best (but a lot worse than the JPHMM method), the simple Pathlet HMM next best, and the Apparance HMM worst.

7 Conclusion and Discussion

We have demonstrated the ability to model subtle facial behaviour using a modified HMM that uses a principled probabilistic framework to learn both short-term visual dynamics and longer-term behavioural structure. The results of psychophysical experiments show that sequences generated by the model are indistinguishable in realism from the original training sequences. Although the AAM parameterisation of the training sequence was tuned to a single individual, we do not believe this is a limitation. We have already demonstrated the ability to track reliably with an AAM - our intention here was to ensure high-quality parameterisation, so that the modelling results would run no risk of being confounded by poor parameterisation. As we indicated in the introduction, this is the first phase of a line of research that is intended to lead to interpretation by synthesis and subtle human-computer interaction. The initial results are promising and may be of value in their own right for image synthesis.

References

[1] Matthew Brand and Ken Shan. Voice puppetry. In *SIGGRAPH*, pages 21–28, 1999.

[2] Neill Campbell, Colin Dalton, and David Gibson Barry Thomas. Practical generation of video textures using the auto-regressive process. In *BMVC*, pages 434–443, 2002.

[3] Gareth Edwards, Chris Taylor, and Tim Cootes. Interpreting face images using active appearance models. In *Proc. of the third International Conference on automatic face and gesture recognition*, pages 300–305, 1998.

[4] Andrew Fitzgibbon. Stochastic rigidity: Image registration for nowhere-static scenes. In *Proc. of the Eighth International Conference On Computer Vision*, pages 662–669, 2001.

[5] Tony Jebara and Alex Pentland. Action-reaction learning: Automatic visual analysis and synthesis of interactive behaviour. In *Int. Conference on Vision Systems*, pages 273–292, 1999.

[6] Neil Johnson and David Hogg. Representation and synthesis of behaviour using gaussian mixtures. 20(12):889–894, 2002.

[7] Michael Jones and Tomaso Poggio. Multidimensional morphable models: A framework for representing and matching object classes. *Int. Journal of Computer Vision*, 2(29):107–131, 1998.

[8] Donald Morrison. *Multivariate Statistical Methods*. Probability and Statistics. Second edition.

[9] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–286, 1989.

[10] Arno Schödl, Richard Szeliski, David Salesin, and Irfan Essa. Video textures. In *SIGGRAPH*, pages 489–498, 2000.

[11] Hedvig Sidenbladh, Michael Black, and Leonid Sigal. Human motion for synthesis and tracking. In *European Conference on Computer Vision*, pages 784–800, 2002.

[12] Thad Starner and Alex Pentland. Visual recognition of american sign language using hidden markov models. In *Int. Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[13] Junji Yamoto, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992.