Modelling Facial Behaviours

F. Bettinger, T. F. Cootes and C. J. Taylor
University of Manchester
Division of Imaging Science and Biomedical Engineering,
Stopford Building, Oxford Road,
Manchester M13 9PT, U.K.
franck.bettinger@cs.man.ac.uk
{tim.cootes,chris.taylor}@man.ac.uk

Abstract

We consider the problem of learning how a person's face behaves in a long video sequence, with the aim of synthesising convincing sequences demonstrating the same behaviours. We describe a novel approach to segment a sequence into short sections, each representing a distinct action (or a part of an action). These sections are grouped and a model of the variability of the action learnt. A variable length Markov model is trained on the sequence of such actions to learn the temporal relationships. The result is a system that can generate realistic sequences of an individual face.

Keywords: human computer interface, active appearance model, facial behaviour, variable length Markov model.

1 Introduction

We seek to develop a system which can model both the appearance and behaviour of a person's face. We would like to be able to present the system with a sufficiently long training sequence of an individual speaking, moving their head and changing expression, and have the system learn a model capable of simulating their behaviour. Such a system would be useful for many applications from computer games to the generation of believable avatars for human-computer interaction. This paper describes a prototype of such a system and demonstrates its performance at learning relatively simple facial behaviours.

In this paper we will concentrate on relatively low-level behaviour (how a person tends to shake their head or the particular way they smile) rather than more high-level behaviours (such as when they smile or the order in which they tend to perform actions). We assume these low-level behaviours are characterised by relatively short time scales and are repeated sufficiently often in a training sequence that we can recognise them and model their variability. Implicit in the work is the assumption that people do not repeat any action exactly (no-one smiles the same way twice), but that it is possible to learn a distribution representing the variations on a particular action that an individual tends to make.

We model the appearance of the individual using a statistical appearance model that combines shape and texture [5], and assume that the input sequence can be tracked sufficiently accurately (in practice using an active appearance model [4]). The sequence can then be represented as a series of points forming a trajectory through the parameter space of the appearance model. A challenging step is then to analyse this trajectory, automatically breaking it down into sub-units which correspond to distinct actions, and to model these actions. We present a novel approach to this, in which we locate nodes in space at points of high density and use these to split the trajectory into segments, which are then grouped and the groups modelled. A variable length Markov model is then trained to learn the relationships between the groups. This allows us to synthesise novel paths through the groups and thus novel sequences which capture the behaviour observed in the training set.

In the following we review related work, describe the system in more detail and show the results of experiments.

2 Related past work

Bregler, in [2], uses a hierarchical framework to recognise human dynamics. His framework can be decomposed into four steps: the raw sequence, a model of movement using a mixture of Gaussians, a model of linear dynamics and a model of complex movements using a hidden Markov model. He highlighted the need of high level information for a correct model of behaviour.

Brand *et al.*, in [1], describes a model of interaction called coupled hidden Markov model. Different behaviours are encoded using separate states for two interlocutors. Each state depends on all the previous states, that is the previous states of both interlocutors. He develops an efficient learning algorithm and shows that this model outperforms classical models such as hidden Markov models.

In [8], shapes are approximated by splines. The parameters controlling those splines as well as their speed are first clustered into prototype vectors using a competitive learning neural network. A compressed sequence derived from the prototype vector sequence is learnt using a Markov chain. A cubic Hermite interpolation is used along with the learnt Markov chain to recover the temporal structure of the sequence before compression and to extrapolate a behaviour. Furthermore, for generation purposes, a single hypothesis propagation and a maximum likelihood framework are described. During the generation, states of the Markov chain are chosen according to the state of the shape of a tracked person. This can allow generation of a shape of a virtual partner driven by a tracked real person. In [6], Devin and Hogg added sound and appearance to the framework in order to demonstrate that producing a talking head is possible. [7] introduces the use of variable length Markov model with the prototype vectors to learn the structure of the sequence.

In [10], Walter *et al.* model gestures by groups of trajectory segments. The trajectory segments are extracted by detecting discontinuities in the gesture trajectory. After normalising the trajectory segments, their dimensions are reduced using a principal component analysis. Clusters are then extracted from the component space using an iterative algorithm based on minimum description length. The clusters form atomic gesture components. There is a parallel between groups of trajectory segments and the actions or visual units we want to extract from the video sequence. However our segmentation and

grouping algorithms are both different.

Finally, the experiments of Martin *et al.* [9] suggest that it is possible to recognise face expressions from their trajectories in the appearance parameter space we use in our model. Thus, our model should be able to generate different expressions.

3 Structure of the model

3.1 Introduction

In order to be able to generate video sequences of faces, we first need an underlying model that is able to synthesise a face for each frame. Thanks to its synthesis facility, the active appearance model of Cootes *et al.* [4] is a perfect candidate for this task.

In order to encode each frame from the training sequence, we use a full appearance model that combines shape and texture information. After having computed the mean shape from the training set, the number of parameters of the model is reduced by applying consecutive principal component analysis to both the shape and the texture part of the model. The details of the model are described in [5]. The shape and a shape-free texture are modelled by the set of linear equations:

$$\begin{cases} x = \overline{x} + Q_x c \\ t = \overline{t} + Q_t c \end{cases}$$

where x is a vector describing the shape, t is a vector describing the shape-free texture, Q_x and Q_t are matrices learnt from the training set. $\overline{x}$ and $\overline{t}$ represent the mean shape and mean shape-free texture computed from the training set.

Given a vector of appearance parameters c, the shape x can be computed. A shape-free texture t can be warped to the shape to reconstruct the full appearance of a face.

Each vector from the appearance parameter space represents a face while each facial image can be approximated by a vector in the appearance parameter space. A sequence of a face can be represented by a trajectory in the appearance parameter space. Visual units are therefore sub-trajectories within this trajectory.

Figure 1 shows an overview of the model of facial behaviour. First, the face has to be tracked in the video sequence (1). The active appearance model parameters have then to be deduced from the tracked face ($1 \rightarrow 2$). The trajectory formed by those appearance parameter vectors is then broken into sub-trajectory groups ($2 \rightarrow 3$) and the sequence is now a sequence of sub-trajectory groups (3). The sequence of sub-trajectory groups is learnt ($3 \rightarrow 4$) by a variable length Markov model (4).

In order to generate new trajectories, a sequence of sub-trajectory groups has to be sampled from the variable length Markov model ($4 \rightarrow 3$). A new sub-trajectory has to be sampled from each group in the sequence of sub-trajectory groups ($3 \rightarrow 2$) to give a sequence of sub-trajectories, that is a trajectory (2). Each point in that new trajectory in the appearance parameter space can then be synthesised ($2 \rightarrow 5$) to give a video sequence of faces (5).

3.2 Extracting the sequence of parameters

The active appearance model is able to fit an appearance model onto a face image, by minimising the difference of texture between the synthesis of the model and the image

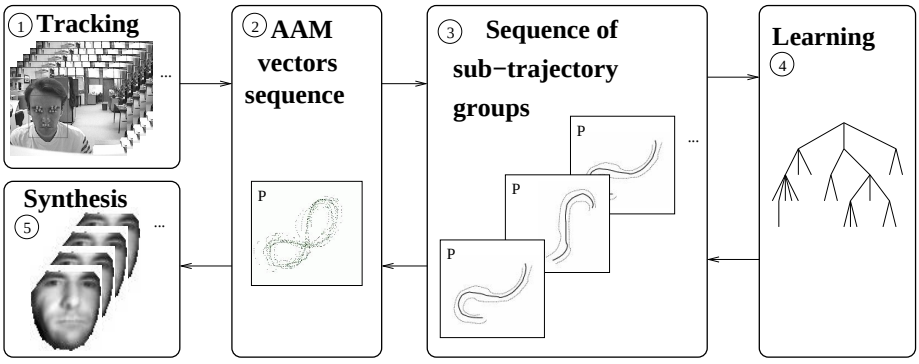


Figure 1: Overview of the components of the model. P is the appearance parameter space. Arrows from left to right represent the learning and arrows from right to left represent the generation.

of interest. As it is a local minimisation, it requires a good first approximation. For each training sequence, the first frame is marked up. In order to get a good approximation for each frame we use the state of the face (pose, scale, position and appearance) from the previous frame as the first approximation for the fitting procedure in the current frame. By comparing the synthesised face and the corresponding pixels in the current frame (using a mean square error on the grey-level pixel values), we can determine whether the fitting procedure has failed or not. If it has failed, a global search is performed.

This semi automatic method has been used to represent an image sequence as a sequence of parameter vectors.

3.3 Segmenting into sub-trajectories

Given a long sequence of points in the parameter space, we want to divide it into sub-trajectories. These sub-trajectories correspond to actions or visual units in the video sequence.

The aim of the segmentation is to extract some nodes that will split the trajectory into several sub-trajectories. The nodes will form the beginnings and ends of the sub-trajectories. The sub-trajectories are computed in order to be grouped later in the process, so similar sub-trajectories should also have similar beginnings and ends respectively. Furthermore, we would like to find the points where different behaviours split or converge together. Finding points of high density in the appearance parameter space is a good way of meeting these requirements.

In order to find the high density points, we use the sample mean shift, described by Comaniciu and Meer [3]. We iteratively modify our current estimate of the local maxima of density by moving to the mean of the n closest points of the current estimate. The process converges to the position of the local maximum density.

We initialise the mean shift algorithm at each point of the trajectory in turn. Running the algorithm to convergence finds all the nearly local maxima in the density estimate. The trajectory points nearest to each local maxima are defined to be the nodes splitting

Figure 2: High density points extracted from a trajectory. The figure on the left represents the sequence of points. The centres of the circles on the figure on the right represents the selected nodes. The four corners where split and merge of trajectories occurs have been correctly identified as nodes.

the full path into sub-trajectories. In practice, we only do this for each k points from the training sequence. This improves efficiency with a negligible effect on the result. Figure 2 shows an example of nodes extracted from a hand drawn trajectory.

3.4 Grouping similar sub-trajectories

3.4.1 The model of a group of sub-trajectories

We model each sub-trajectory using a linear statistical model, assuming a Gaussian distribution. Each sub-trajectory is described by a vector, which is a simple concatenation of the sub-trajectory points. A sub-trajectory group is a set of vectors, on which we can apply a principal component analysis. Each sub-trajectory s is approximated by:

$$s = \overline{s} + Q_s b_s \tag{1}$$

where $\overline{s}$ is a vector representing the mean sub-trajectory of the group, Q_s is a matrix computed by the principal component analysis and describing how the data varies, and b_s is the vector of parameters for that particular sub-trajectory. The probability density function of the set of parameters b_s is modelled by a Gaussian, by computing the mean and variance of b_s for all sub-trajectories s in the group.

In order to be able to perform the principal component analysis on a group of sub-trajectories, it is required that all the sub-trajectories are encoded with the same number of points. Therefore, we interpolate all the sub-trajectories by cubic splines and homogeneously re-sample them to a given number of points.

3.4.2 The grouping algorithm

We experimented with two grouping algorithms. The first is a greedy algorithm. It first assumes that each sub-trajectory given by the segmentation initially forms a group by itself. Let $\mathcal{S}$ be this initial set of groups.

We want to know how to merge groups so that the resulting groups are sufficiently coherent that a Gaussian model is a good representation.

For every pair (g_i, g_j) of elements of $\mathcal{S}$, we compute the variance of the group $g_{i \cup j}$ that is built using the sub-trajectories from both g_i and g_j. We select the pair of groups (g_a, g_b) that gives the lowest variance and merge those two groups to form only one group. We delete g_a and g_b from $\mathcal{S}$ and insert $g_{a \cup b}$ instead.

We iterate the process until we reach a given number of clusters.

Unfortunately this algorithm becomes quite inefficient if we want to process a large number of sub-trajectories (several hundreds in practice). Its complexity is $\mathcal{O}\left(p^4\right)$ where p is the total number of sub-trajectories.

For larger training sets of sub-trajectories, we use the normalised cuts algorithm of Shi and Malik [12]. This computes groups by analysing the eigenvectors of a $p \times p$ similarity matrix. The computation of this matrix is in $\mathcal{O}\left(p^2 S(p)\right)$ where $S(p)$ is the complexity of the similarity measure. The complexity of the solution to the eigenvalue problem usually has a complexity of $\mathcal{O}\left(p^3\right)$ but fortunately, for a sparse matrix, it can be reduced to a lower complexity, using the Lanczos algorithm.

We tried this algorithm with two similarity measures, one is the euclidian distance, and the other one is based on dynamic time warping. Both have a complexity of $\mathcal{O}\left(p\right)$ which brings the complexity of the whole algorithm to $\mathcal{O}\left(p^3\right)$. The results are slightly better for the euclidian distance because the dynamic time warping measure is invariant to transformations such as rotations or translations. The principal component analysis used to compute equation 1 performs poorly if we use a similarity measure which is not strict enough on the allowed transformations.

The results, even if still acceptable, are visually less effective than those from the greedy algorithm. However, the normalised cuts method scales better for a large number of sub-trajectories.

3.5 Learning temporal relationships between groups

We now want to learn the structure of the sequence of sub-trajectory groups. In order to do that, we use a variable length Markov model introduced by Ron *et al.* [11] in the context of learning a sequence of letters in English texts. The idea of this model is to use a memory of length varying with the context. In particular, when the trajectory splits into two trajectories, we need to know where the sub-trajectories are coming from, in order to decide what is the next sub-trajectory to infer. A memory of a long sequence of sub-trajectory groups is needed in that case. On the other hand, if group A is always followed by group B, we need only a short memory (stating that if the current group is A, then B is next).

The probabilities of sequences of sub-trajectory groups are stored in a tree. Each branch of the tree corresponds to a sequence. The tree is constructed recursively by adding sequences. A sequence l is added to the tree if and only if its probability is greater than a previously chosen threshold and adding the sequence l gives a statistically different tree. Two trees are said statistically different if the probability distributions encoded by the trees, weighted by the probability of the sequence l, are different. Thus we need a measure of distance between probability distributions. The Kullback-Leibler measure is usually used with the variable length Markov model. However, by trying the model on the generation on English texts, we found that the Matusita measure gives better results on long texts (about 33% of letters were successfully predicted against 26% for the Kullback-Leibler measure). The Matusita measure is given by:

$$D(p\|q) = \int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \, dx$$

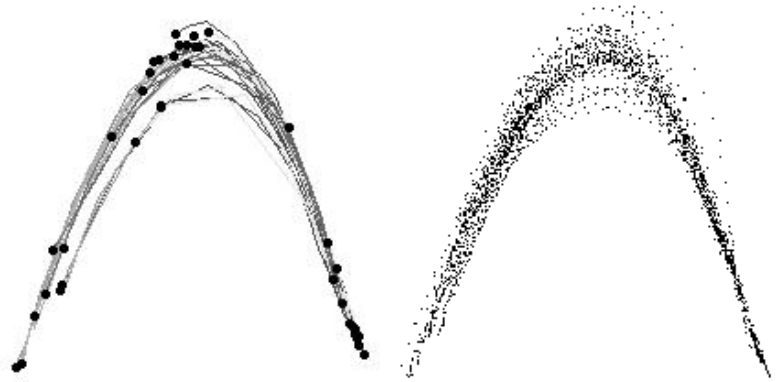
for two probability distributions p and q. $p(x)$ is the probability of observing a group

x given the sequence l. $q(x)$ is the probability of observing a group x given the subsequence of l that represents a shorter memory already encoded in the tree. The statistical difference tells us whether there is any advantage in using longer memory for the current context. The algorithm stops when every sequence of length less than a predefined threshold has been inspected.

3.6 Generating new sequences

A new video sequence of faces can be generated from the model as follows. First, given an history of generated sub-trajectory groups, one can find the longest memory encoded in the variable length Markov model tree. Thus the probability of generating a new group can be read directly from the tree, if it is encoded in the tree. The probabilities not encoded in the tree are small probabilities, that we can approximate by a uniform distribution. After having fetched the probabilities of generation of each sub-trajectory groups, we sample from this set of probabilities to generate the next sub-trajectory group. We then generate new parameters by sampling from a Gaussian distribution. The new sub-trajectory can then be generated given equation 1. All the sub-trajectories generated are then concatenated. This gives a sequence of appearance parameters. The video sequence can then be generated by synthesising those parameters into a face as described in section 3.1.

4 Experimental results



(a) Extraction of nodes (b) Generated new sequence

Figure 3: Figure 3(a) represents the extraction of nodes from the training video sequence. The plan of the graph represents the two first components of the seven components of the appearance parameter space. The points representing the sequence are linked by lines if they follow each other. The extracted nodes are represented by black circles. Figure 3(b) represents the points of a new sequence generated using the model.

Figure 4: The six groups obtained after the grouping of sub-trajectories. Groups 1 and 2 look similar because one group models trajectories that go from left to right while the other one models trajectories that go from right to left. The same applies for groups 3 and 4. The group 5 corresponds to several really small trajectories.

We have applied the algorithm to a sequence of a person repeatedly shaking their head. Frames extracted from the training sequence can be seen on figure 5(a). Those frames are the synthesis of the appearance parameters sequence after tracking. The whole sequence has 317 frames. Each face is encoded using seven parameters including the scale, pose and position. The sequence of the two first components is represented on figure 3(a) along with the nodes extracted from the sequence. The nodes are extracted using $n = 10$ and $k = 20$.

Figure 4 shows the result of the grouping algorithm if we ask for 6 clusters. Figure 3(b) shows an example of trajectory that can be generated using those 6 clusters and a variable length Markov model that has been trained with a threshold of 0.001 for both the probabilities stored and the statistical difference. The synthesis of a generated sequence can be found on figure 5(b).

5 Conclusions and future work

We have presented a generative model of visual facial behaviour that is based on the assumption that people repeat facial expressions over time. It has been shown that this model is able to reproduce simple behaviours.

A novel decomposition of sequences into visual units allows a higher level of abstraction than other models that are based on sequences of frames. The use of variable length Markov model allows us to efficiently encode history of visual units in long sequences. Generation of new sequences can be done thanks to the combination of the generative features of both the variable length Markov model and the appearance model. Furthermore, the use of variable length Markov model for learning helps us to better understand what is going on with the model. There are no hidden variables or hidden states.

However, the transition between visual units has to be smoothed. For the time being, some jumps are perceptible in the generated sequences. Another improvement that can be done is to modify the model so that it handles outliers and timings in a better way.

In our future work, we plan to use two active appearance models with the same framework in order to model interactions between two persons speaking together in an interview type scenario in a similar manner to [6].

References

[1] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *IEEE CVPR97*, 1996.

[2] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.

[3] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Seventh International Conference on Computer Vision*, pages 1197–1203, 1999.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H.Burkhardt & B. Neumann Ed's, editor, *European Conference on Computer Vision*, volume 2, pages 484–498. Springer, 1998.

[5] T. F. Cootes and C. J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Proc. SPIE Medical Imaging*, 2001.

[6] Vincent E. Devin and David C. Hogg. Reactive memories: An interactive talking-head. In *British Machine Vision Conference*, pages 603–612, September 2001.

[7] Aphrodite Galata, Neil Johnson, and David Hogg. Learning variable-length Markov models of behavior. *Computer Vision and Image Understanding: CVIU*, 81(3):398–413, March 2001.

[8] N. Johnson, A. Galata, and D. Hogg. The acquisition and use of interaction behaviour models. In IEEE Computer Society Press., editor, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR'98*, pages 866–871, 1998.

[9] J. Martin, D. Hall, and J. L. Crowley. Statistical gesture recognition through modelling of parameter trajectories. In *Third Gesture Workshop*, 1999.

[10] Alexandra Psarrou Michael Walter and Shaogang Gong. Data driven gesture model acquisition using minimum description length. In *British Machine Vision Conference*, pages 673–683, September 2001.

[11] Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 176–183. Morgan Kaufmann Publishers, Inc., 1994.

[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Conf. Computer Vision and Pattern Recognition*, June 1997.

(a) Training sequence of a face gesturing ”no”

(b) Generated sequence

Figure 5: Figure 5(a) represents some frames extracted from the video sequence used to train the model, while figure 5(b) represents some frames extracted from the sequence generated by the model.