A model of facial behaviour

Abstract

We wish to model the way in which faces move in video sequences. We represent facial behaviour as a sequence of short actions. Each action is a sample from a statistical model representing the variability in the way it is performed. The ordering of actions is defined using a variable length Markov model. Action models and variable length Markov model are trained from a long (20000 frames) video sequence of a talking face. We propose a novel method of quantitatively evaluating the quality of the synthesis by measuring overlaps of parameter histograms. We apply this method to compare our technique with an alternative model that uses an autoregressive process.

1. Introduction

This paper describes a prototype of a facial behaviour learning system and demonstrates its performance at learning facial behaviours by comparing the results with an alternative model proposed by Campbell *et al.* [3].

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.

Our model is designed for analysing 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.

In the following we review related work, describe our system in more detail and show how we compare models of facial behaviour. The results of the comparison between the two models of facial behaviour are given in section 5.2.

2. Previous 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.

In [9], 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 [7], Devin and Hogg added sound and appearance to the framework in order to demonstrate that producing a talking head is possible. [8] introduces the use of variable length Markov model with the prototype vectors to learn the structure of the sequence.

In [15], 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.

Ng and Gong [11] use the Levenshtein distance along with an unsupervised version of the Normalised Cut algorithm [14] to group gesture trajectories. Unfortunately, we can still spot out outliers in some resulting groups, which leads to problematic modelling of those groups.

Finally, the experiments of Martin *et al.* [10] suggest that it is possible to recognise facial expressions from their trajectories in the appearance parameter space we use in our model.

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.* [5] 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 [6]. The shape and a shape-free texture are modelled by the set of linear equations:

$$\begin{cases} x = \overline{x} + Q_x c \\ t = \overline{t} + Q_t c \end{cases} \tag{1}$$

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 of interest. As it is a local minimisation, it requires a good first approximation. For each training sequence, the first frame is annotated. 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. In practice, only a few global searches need to be done in order to track the face accurately.

This 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

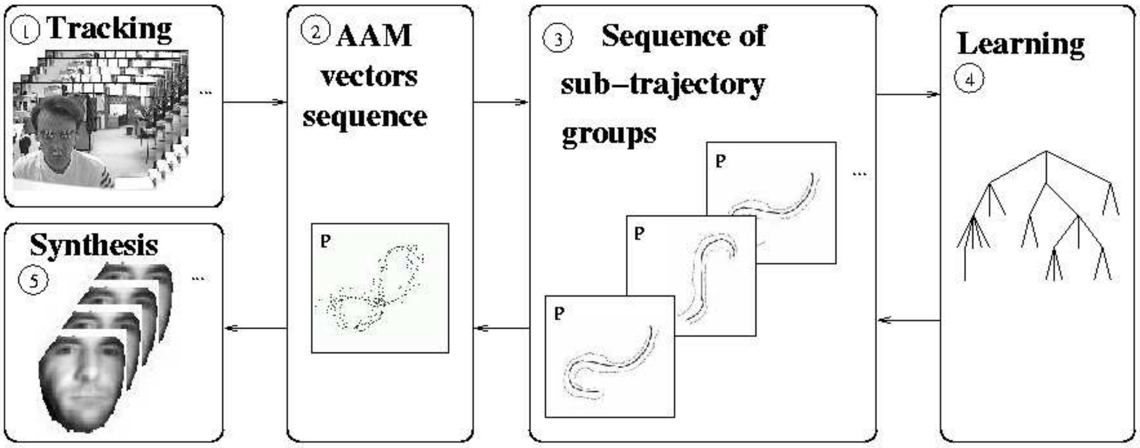


Figure 1: Overview of the components of the model. P is the appearance parameter space. Arrows from left to right represent the learning and arrows from right to left represent the generation. Each face from the frames of the original video sequence corresponds to a point in the space P. The trajectory of the points in P is modelled by a sequence of sub-trajectory groups. The temporal relationship of the groups is learnt by a variable length Markov model.

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 [4]. 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 the full path into sub-trajectories. In practice, we only do this for every k points in the training sequence. This improves efficiency with a negligible effect on the result.

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. We want each sub-trajectory to be 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. 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 resample them to a given number of points. The resulting points after resampling do not have the same timings as the original ones. We need to keep track of this information. The vector used to describe a sub-trajectory is therefore extended with the timing information. Each extra coordinate represents the ratio of the time necessary to reach a point from the first point of the sub-trajectory with respect to the total time of the sub-trajectory. Each sub-trajectory s, along with the corresponding timings δ, is approximated by:

$$\begin{pmatrix} s \\ \delta \end{pmatrix} = \begin{pmatrix} \overline{s} \\ \overline{\delta} \end{pmatrix} + Q_{s,\delta} \begin{pmatrix} b_s \\ b_\delta \end{pmatrix} + R_{s,\delta} \qquad (2)$$

where $(\overline{s}, \overline{\delta})$ is a vector representing the mean sub-trajectory of the group along with this timings, $Q_{s,\delta}$ is a matrix computed by the principal component analysis and describing how the data varies, $R_{s,\delta}$ are the residuals and (b_s, b_δ) is the vector of parameters for that particular sub-trajectory. The probability density function of the set of parameters (b_s, b_δ) is modelled by a Gaussian, by computing the mean and variance of (b_s, b_δ) for all sub-trajectories (s, δ) in the group.

Generation of sub-trajectories is therefore a simple sampling from the distribution of (b_s, b_δ). Applying equation 2 gives a sub-trajectory s which is then re-sampled using cubic splines, and timing information δ.

3.4.2 The grouping algorithm

The grouping algorithm first assumes that each sub-trajectory given by the segmentation initially forms a group by itself. The timings δ are not taken into account for the grouping. 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.

More details on the algorithm and its performance can be found in [1].

3.5 Learning temporal relationships between groups

In order to learn the structure of the sequence of sub-trajectory groups, we use a variable length Markov model introduced by Ron *et al.* [13]. More details are given in [1]. The algorithm basically constructs a decision tree that stores probabilities of common sub-trajectory group sequences.

3.6 Generating new sequences

A new video sequence of faces can be generated from the model as follows. First, given a 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 sequences not encoded in the tree have small probabilities, that we can approximate by an 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 as described in section 3.4.1. A linear model is chosen for the residuals $R_{s,\delta}$ so that the beginning of the generated sub-trajectory matches the end of the previous generated sub-trajectory to avoid perceptible jumps in the generated video. 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.

An example of generated trajectory is given by Figure 4(c). The three dimensions represent the three first modes of variation of the active appearance model used. Another example of generated video can be seen on Figure 3. The original video sequence is shown on Figure 2.

Figure 2: Frames taken every 4 seconds from the original long video sequence.

Figure 3: Frames taken every 4 seconds from the video sequence generated with our model.

4. Campbell's model

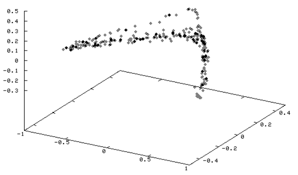
In [3], Campbell *et al.* introduce another way of generating video sequences of faces based on an existing video clip without direct reuse of the original frames. They encode frames from the original sequence in a way similar to our method. An active appearance model is used to model the face through the video sequence and a trajectory is obtained in the appearance parameter space. This trajectory is then used to train a second order autoregressive process.

An autoregressive process is simply trying to predict the position of a point x_k in the appearance parameter space, given the two previous points x_{k-1} and x_{k-2} where k represents the frame number. This prediction is produced by the equation:

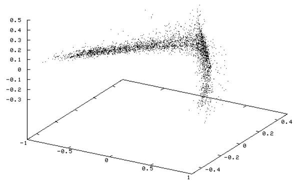
$$x_k - \overline{x} = A_2\left(x_{k-2} - \overline{x}\right) + A_1\left(x_{k-1} - \overline{x}\right) + B_0 w_k \quad (3)$$

where $\overline{x}$ is the limit of the mean value of x_k as k tend to infinity, w_k is a white noise ($w_k \sim \mathcal{N}(0,1)$), A_2, A_1 and B_0 are parameter matrices. $\overline{x}$, A_2, A_1 and B_0 can be learnt from the original data set. The learning method used in this work is due to Reynard *et al.* and is described in [12].

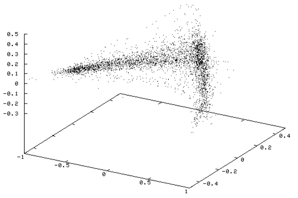
Given two initial points in the parameter space, a new trajectory can be generated by applying equation 3 repetitively. This new trajectory in the parameter space can then be synthesised back to a video sequence of a face in a similar way to ours. Figure 5 shows an example of a video clip generated by Campbell's model.



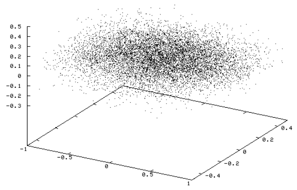
(a) Training sequence of a face gesturing "no"



(b) Generated sequence with our model without using the residuals



(c) Generated sequence with our model



(d) Generated sequence with Campbell's model

Figure 4: Figure 4(a) represents points on the trajectory that correspond to the original video sequence. Figure 4(b) represents points on the trajectory generated by our model if the residuals used are set to zero. Figure 4(c) represents the equivalent with a linear model for residuals. Figure 4(d) represents points on the trajectory generated by Campbell's model.

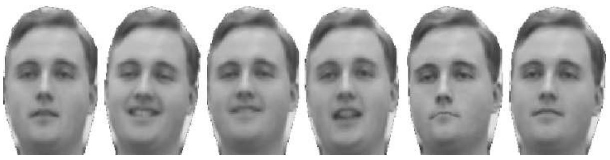


Figure 5: Frames taken every 4 seconds from the video sequence generated with Campbell's model.

5. Comparison of the two models

5.1 Method of comparison

In order to compare the two models, we need some measure of behavioural similarity between two video sequences. Our approach is to compare the distribution of the generated points in the parameter space with the distribution of points extracted from the original video sequence.

We construct two dimensional histograms to approximate the distribution of points in the parameter space for each pair of dimensions. In order to compare the original and generated sequences of points using histograms, particular care has to be taken on the selection of the bin width used to compute those histograms. Indeed, a too large bin width will act as a smoothing effect on the original data while a too small bin width will result in an over-fitting of the data. In order to solve this problem, and for reproducibility of the comparison method, we used Scott's rule to select the bin size [16]. The bin size for each dimension is given by the formula:

$$h = 3.5\sigma N^{-\frac{1}{3}} \tag{4}$$

where N is the total number of points in the sequence and σ is the standard deviation of the data computed with respect to the selected dimension.

The two dimensional histograms of a reference and a generated set of points are then compared using the mean of the Bhattacharyya overlap computed for each pair of dimensions. The standard error can also be computed to represent our confidence in the result. This measure represents how close the point distributions of the generated and the reference sequences are. A value of 1 corresponds to a perfect match of the distributions. A value of 0 corresponds to two totally different distributions.

The facial behaviour models are assessed with respect to the original video sequence used to create those models. The distributions of points in the appearance parameter space should match with the original one if the model performs its task correctly.

5.2 Experimental results and discussion

6. Summary and Conclusions

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 measure of similarity between video sequences of facial behaviour has been developed and used to assess the model as well as an alternative based on autoregressive process. Our model performs better on highly structured video sequences such as nodding where expressions are repeated over time. It also produces a better distribution of speed. The model is capable of synthesising convincing facial behaviour. We are currently investigating extending the work to include interactive behaviour between two people as they talk.

References

[1] Anonymous. Anonymous title. In *Anonymous book title*, Anonymous year.

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

[3] Neill W. Campbell, Colin Dalton, David Gibson, and Barry Thomas. Practical generation of video textures using the

Sequence	Model	Measure	Value
No : 	ARP model	position :	$\mu = 0.89798, \sigma = 0.01022$
		speed :	$\mu = 0.86338, \sigma = 0.00785$
	w/o residuals	**position :**	$\mu = 0.96500, \sigma = 0.00319$
		speed :	$\mu = 0.90980, \sigma = 0.00434$
	with residuals	**position :**	$\mu = 0.94379, \sigma = 0.00429$
		speed :	$\mu = 0.89618, \sigma = 0.00527$
Expressions : 	ARP model	position :	$\mu = 0.82789, \sigma = 0.01112$
		speed :	$\mu = 0.78151, \sigma = 0.01098$
	w/o residuals	position :	$\mu = 0.82859, \sigma = 0.00898$
		speed :	$\mu = 0.90124, \sigma = 0.00516$
	with residuals	**position :**	$\mu = 0.86494, \sigma = 0.00611$
		speed :	$\mu = 0.89734, \sigma = 0.00482$
Dialog : 	ARP model	position :	$\mu = 0.89567, \sigma = 0.00341$
		speed :	$\mu = 0.86978, \sigma = 0.00198$
	w/o residuals	**position :**	$\mu = 0.93054, \sigma = 0.00144$
		speed :	$\mu = 0.86372, \sigma = 0.00208$
	with residuals	position :	$\mu = 0.81251, \sigma = 0.00554$
		speed :	$\mu = 0.90309, \sigma = 0.00236$

Table 1: Comparison of the autoregressive process with our model, with and without the residual modelling.

auto-regressive process. In Paul L. Rosin and David Marshall, editors, *British Machive Vision Conference*, pages 434–443, September 2002.

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

[5] 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.

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

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

[8] 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.

[9] N. Johnson, A. Galata, and D. Hogg. The acquisition and use of interaction behaviour models. In IEEE Computer Society Press., editor, *CVPR*, pages 866–871, 1998.

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

[11] Jeffrey Ng and Shaogang Gong. Learning intrinsic video content using Levenshtein distance in graph partitioning. *Lecture Notes in Computer Science*, 2353:670–684, 2002.

[12] David Reynard, Andrew Wildenberg, Andrew Blake, and John A. Marchant. Learning dynamics of complex motions from image sequences. In *ECCV (1)*, pages 357–368, 1996.

[13] 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.

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

[15] M. Walter, A. Psarrou, and S. Gong. Auto clustering for unsupervised learning of atomic gesture components using minimum description length. In *ICCV-RATFG-RTS*, pages 157–162, 2001.

[16] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59, February 1997.