Speech Driven Facial Animation using a Hidden Markov Coarticulation Model

Darren Cosker, Dave Marshall, Paul. L. Rosin and Yulia Hicks
School of Computer Science, Cardiff University
PO Box 916, Cardiff CF24 3XF, U.K.
D.P.Cosker, Dave.Marshall, Paul.Rosin, Y.A.Hicks@cs.cf.ac.uk

Abstract

We present a hierarchical image based facial model which is driven from speech. It incorporates a novel modelling and synthesis algorithm for learning and producing coarticulated mouth animation We demonstrate using the hierarchical model how animation of the entire face may be created solely from animations of the mouth and how colour may be reincorporated and reproduced compactly without explicitly being modelled. We then consider and evaluate methods of merging animations from several different facial areas for delivery of output.

1. Introduction

We describe a hierarchical image-based facial animation system capable of producing coarticulted mouth animation given audio input alone. The system requires a short video of a speaker to learn mouth/speech coarticulation and once trained is fully automatic. We extend previous work [3] where we produced mouth animations using a speech-appearance based joint statistical model. The limitation of this method was that coarticulation was not fully modelled, often leading to degraded animation.

The model described in [3] divides the face hierarchically into sub-facial areas, each of which is animated separately from speech and merged in the final output using warping methods. In this paper we introduce several advances where animation need only be produced for leaf nodes which drive animations for their parents. We demonstrate this by describing how realistic facial animation may be produced solely from an animation of the mouth. We also introduce a novel method of producing colour appearance model based outputs using this framework.

Figure 1 shows an example hierarchical model. Each node of the hierarchy contains a grey-level/luminance based appearance model [2] of a specific facial area along with an image *hue* and *saturation* Principle Components Analysis (PCA) model. Leaf nodes in the hierarchy produce streams of appearance parameters driven by speech input. Parent nodes then use these appearance parameters to generate their own model-specific parameters for grey level, hue and saturation to yield a colour output. Producing image-based

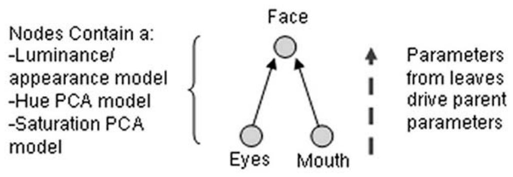
Nodes Contain a:
-Luminance/ appearance model
-Hue PCA model
-Saturation PCA model
Face
Eyes
Mouth
Parameters from leaves drive parent parameters

Figure 1. Hierarchical overview.

facial animation using this framework has several advantages in terms of modelling and compactness. Building appearance models of specific facial areas ensures that specificity and generalisation are improved more than if the entire face is modelled. Compactness and complexity is also improved since colour data (excluding luminiance) need not be generated separately for animation nor explicitly included in an appearance model.

2. Training and initialisation

For each node in our hierarchical model we require that an appearance model of that facial area be built. We initially collect data for the entire face and extract sub-facial data as the hierarchy requires. We first record a few minutes of video of a speaker reading a text or reciting a story at 25fps with mono audio sampled at 32KHz. The subject is recorded front-on with as little out of plane head movement as possible. We record the subject's voice using the on camera microphone. Each image in the video corpus is then annotated with landmarks highlighting key facial areas. We achieve this semi-automatically using the Downhill Simplex tracker described in [5]. Sub-facial appearance models are then built by extracting image and landmark data at specific facial areas. Once this data has been extracted we follow the procedure for building appearance models as outlined in [2]. The only major differences in our technique are

1) we convert our images into YIQ colour space and use luminance data to build our appearance models, and 2) we do not align our shape data with respect to scale. The reason for the second point is that we regard scale in the mouth as extra shape information. Figure 2 shows example annotated images used for our model. After data collection for a node/sub-facial area we represent our luminance/shape appearance model along with the hue and saturation PCA models respectively as

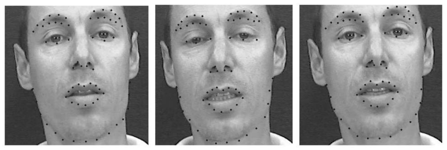

Figure 2. Data initialisation.

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_x \mathbf{W}_x^{-1} \mathbf{Q}_x \mathbf{c} \tag{1}$$

$$\mathbf{g}_{lum} = \bar{\mathbf{g}}_{lum} + \mathbf{P}_{lum} \mathbf{Q}_{lum} \mathbf{c} \tag{2}$$

$$\mathbf{g}_i = \bar{\mathbf{g}}_i + \mathbf{P}_i \mathbf{b}_i \qquad i = \{hue, sat\} \tag{3}$$

where matrices **P** and **Q** represent eigenvectors, vectors $\bar{\mathbf{g}}$ and $\bar{\mathbf{x}}$ represent model means, vectors **x** and **g** represent shape and texture data, vectors **b** and **c** represent eigenvector weights, and matrix $\mathbf{W}_x^{-1}$ is a shape-to-texture parameter scale matrix. Note we use **c** to represent an appearance parameter. For a fuller description of these models see [2].

2.1. Speech feature acquisition

To obtain robust and uncorrelated speech features we follow common speech recognition techniques. Given a raw audio track extracted from the captured training corpus we perform Mel-Cepstral analysis using 20ms Hamming windows and retain 12 coefficients[4]. We then perform Cepstral Mean Normalisation (CMN) on this set of coefficients[4].

Given this set of coefficient vectors we construct a linear PCA speech model and project each vector through the model, yielding new PCA weight vectors $\mathbf{b}_S$ corresponding to coefficient vectors. To account for the 50Hz speech sampling rate and the 25Hz frame rate we linearly interpolate the mouth appearance training set. This gives us one-to-one correspondences between our speech and appearance training data with which to build our model.

3. Mouth Animation

In this section we present a means of generating mouth animation at a leaf node. The process may also be reused for other leaves or may be used stand-alone with any appearance model. The most important and difficult problem for mouth animation is that of coarticulation. Deciding which mouth to output at time *t* given the current frame of speech depends both on past and forthcoming observations of speech and visemes.

This process may be considered as Markovian and modelled with a Hidden Markov Model(HMM)[7]. A HMM is a doubly embedded stochastic process where one process is *hidden* and only observed through a second one which produces a set of observations. The mouth appearance parameter training set is modelled using a standard gaussian output HMM where the hidden process relates to selection of different mouth groups (HMM states). What is required is a means of deriving this hidden process from a speech input. Brand's solution is to build a second identical HMM except with means and covariances calculated from the speech training set[1]. The hidden state sequence of the appearance HMM may then be calculated using the speech HMM, a new speech input and the Viterbi algorithm.

3.1. Dual HMM Construction

Given a mouth appearance parameter training set we build a HMM with V states and define it as $\lambda_A = (T_A, B_A, \pi_A)$ where T_A is our state transition probability distribution, B_A is our observation symbol probability distribution and π_A is our initial state distribution. In construction of our HMM we also obtain the matrix γ which tells us the probability of being in state *k* at time *t*. Using the matrix γ we build a second HMM $\lambda_S = (T_A, B_A, \pi_A)$ with means and covariances calculated using the speech training parameter set. For details on this process refer to [7].

This dual-HMM framework conveniently allows us to estimate a hidden state sequence in λ_A given any speech observation using λ_S and the Viterbi algorithm [7]. However the problem of output still remains unsolved since we still do not know which appearance parameter to display given a probable hidden state sequence in λ_A. We begin by rigidly allocating appearance parameters to states in λ_A by calculating the Mahalanobis distance between each parameter and a state and assigning it to the closest. Using this above process we are also able to simultaneously classify speech parameters into states. We record these mappings for use in Section 3.2.

Given a probable state sequence through λ_A the problem is now simplified to choosing an output parameter associated with a state at each time *t*.

3.2. Estimating output parameters

To solve the problem of choosing an appearance parameter to display for each state we proceed as follows. We define a trellis data structure where each column represents the appearance parameters allocated to state Q_t and Q is the set of states resulting from the Viterbi algorithm step. Note that the length of each column may therefore vary and that each parameter also corresponds to a speech parameter from the training set (see Section 3.1). For each appearance parameter in each column we assign an error which defines the cost of observing that appearance parameter. Once we have defined costs for each parameter we work backward through the trellis and choose appearance parameters with the lowest cost for each time *t*. Figure 3 illustrates an example of an initial trellis structure. We calculate errors at *t*=1 using

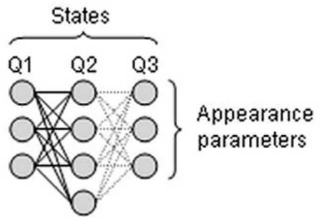

Figure 3. Trellis data structure.

$$E(\mathbf{c}_j, Q_1) = ((\mathbf{b}_S^j - \mathbf{b}_{input}^1)^T \Sigma_{Q_1}^S (\mathbf{b}_S^j - \mathbf{b}_{input}^1)) \quad j = 1, \ldots, p \qquad (4)$$

where $E(\mathbf{c}_j, Q_1)$ is the error for appearance parameter $\mathbf{c}_j$ given state Q_1, *p* is the number of appearance parameters in column Q_1, $\mathbf{b}_{input}^1$ is the input speech parameter observed at *t*=1, $\mathbf{b}_S^j$ is the speech parameter associated with a corresponding appearance parameter and $\Sigma_{Q_1}^S$ is the speech covariance matrix for state Q_1.

To calculate the errors for parameters at $t = 2, \ldots, M$, where *M* is the length of the input speech observation we use

$$E(\mathbf{c}_j, Q_t) = \left[\sum_{i=1}^{r} (\mathbf{c}_{t-1}^i - \mathbf{c}_t^j)^T \Sigma_{Q_t}^A (\mathbf{c}_{t-1}^i - \mathbf{c}_t^j) \right] \times ((\mathbf{b}_S^j - \mathbf{b}_{input}^t)^T \Sigma_{Q_t}^S (\mathbf{b}_S^j - \mathbf{b}_{input}^t)) \qquad (5)$$

where $j = 1, \ldots, p$ and *p* is the number of cells in group Q_t, *r* is the number of cells in group Q_{t-1}, $\Sigma_{Q_t}^A$ is the covariance matrix in state Q_t of the appearance HMM, $\mathbf{c}_t^j$ is the appearance parameter for cell *j* at time *t* and $\mathbf{c}_{t-1}^i$ is the appearance parameter for cell *i* at time *t-1*. Calculation for the score in a cell is illustrated in Figure 4. The procedure essentially allocates errors based on the distances between an appearance parameter in a column at time *t* and parameters at time *t-1*, while taking into account how close the input speech matches the speech parameter for that appearance parameter. Once errors are calculated for each appearance parameter we travel backwards through the trellis and choose the appearance parameter in each column with the lowest error at time *t*. This gives an output trajectory of appearance parameters for mouth animation. Since we are essentially reordering training set mouth parameters we also retrieve hue and saturation training vectors for output. We use these for colour synthesis in the following sections.

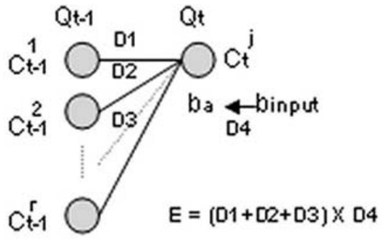

Figure 4. Errors for *t* $= 1, \ldots,$ *M*.

4. Trajectory post-processing

Although the trellis algorithm considers neighboring frame similarity it can not guarantee it, since it is also influenced by speech distances. To ensure smooth animations we implement a weighted averaging filter using *confidence* values. These are created for each time *t* by normalising values in a trellis column, subtracting from 1 and renormalising giving a value between 0 and 1 indicating the level of confidence in an appearance parameter. By using these values in a weighted mean window we suppress appearance parameters with low confidence values and amplify parameters with high confidence.

5. Driving facial animation

Given a trajectory of leaf node appearance parameters we may automatically generate parameters for its parents. Formally we seek a facial appearance parameter at time *t* which when synthesised contains a mouth identical to the corresponding synthesised mouth appearance parameter. Similarly we must also create appropriate hue and saturation information for the face.

5.1. Downhill Simplex Search

The first method we consider is a search for the face appearance parameter which minimises the function $E = \mathbf{g}_{lm} - \mathbf{g}_{pm}$ using the Downhill Simplex Minimisation(DSM) algorithm [6] where $\mathbf{g}_{lm}$ is the luminance generated from the mouth appearance trajectory and $\mathbf{g}_{pm}$ is the luminance extracted from the mouth in the face texture. At each iteration we synthesise the mouth texture from the mouth appearance trajectory at time *t* and warp it to the mouth in the face calculated from the current face appearance parameter. The error E is then recalculated and a new face appearance parameter is chosen using the DSM algorithm.

The algorithm terminates once E is below a threshold or the maximum number of iterations is exceeded. We then repeat the process to find hue and saturation values for the face using equation 3 and parameters $\mathbf{b}_{hue}$ and $\mathbf{b}_{sat}$ as opposed to face appearance parameters.

5.2. Parent model approximation

The second method we consider is perhaps the most accurate and the simplest. Given a generated mouth texture we warp it to over the mouth in the mean face texture. We then substitute the generated mouth shape information into the mouth information contained in the mean face shape. Using the face appearance model (equations 1 and 2) we then estimate a facial appearance parameter. By using facial appearance space in this way we find an estimation of a new appearance parameter which contains the merged face and mouth data. Re-synthesis of a facial image using this parameter results in face image and shape data which accurately contains the desired mouth information. As in Section 5.1 we repeat the process for hue and saturation vectors using equation 3.

6. Evaluation

We recorded a subject speaking for approximately 3 minutes, front-on and with minimal out-of-plane head rotation. Using this data we built a hierarchical model with nodes for the face and the mouth. We then recorded the speaker for another 3 minutes and used this new audio to generate appearance trajectories for the mouth leaf node. Example animations may be found at *http://www.cs.cf.ac.uk/user/D.P.Cosker/demos.html*.

To compare the effectiveness of our reconstruction techniques (Sections 5.1 and 5.2) we took 100 ground truth face appearance parameters from the training set and generated two sets of new facial appearance parameters using 100 ground truth mouth appearance parameters. We evaluated both methods using a Mahalanobis Distance measure between ground and synthesised appearance parameters. Over 100 frames we found the average Mahalanobis distance for the DSM method to be 32.81 and for the parent approximation method to be 31.15 with variances of 168.01 and 118.87 respectively. Subjectively we found the DSM method to produce the poorest animations due to a tendency for the search to result in local minimas.

Figure 5 shows an example animation trajectory using our mouth animation algorithm with the parent approximation method. We judged the mouth animations to be strongly lip-synched to input audio and video-realistic. One drawback however, occurs when the speech rate of the speaker is too fast, resulting in missed mouth articulation for artifacts such as fricatives and plosives. We believe that this may be improved using smaller training set analysis windows (i.e. > 50Hz sampling).

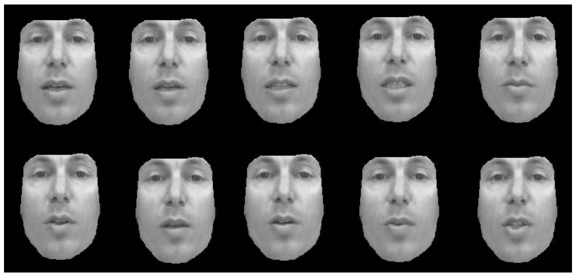

Figure 5. Facial output example using parent approximation.

7. Conclusions

We have presented a highly compact hierarchical framework for producing realistic facial animation from speech. The framework synthesises full facial animation using only data from smaller facial areas such as the mouth to generate colour animations for the whole face. We have compared novel DSM and parent approximation methods for adding sub-facial animations to full facial models. Using the parent approximation method we have demonstrated how the model may generate video-realistic output animations. In future work we aim to further exploit the hierarchical frame work to provide animations for more facial areas.

References

[1] M. Brand. Voice puppetry. In *Proc. ACM SIGGRAPH*, pages 21–28, 1999.

[2] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–684, 2001.

[3] D. Cosker, A. D. Marshall, P. L. Rosin, and Y. A. Hicks. Video realistic talking heads using hierarchical non-linear speech-appearance models. In *Proc. of Mirage*, pages 20–27, 2003.

[4] J. Deller, J. Proakis, and J. Hansen. *Discrete-Time Processing of Speech Signals*. 1993.

[5] J. Luettin and N. A. Thacker. Speechreading using probabilistic models. *CVIU*, 65(2):163–178, 1997.

[6] J. A. Nelder and R. Mead. A simplex method for function minimisation. *Computer Journal*, 7:308–313, 1965.

[7] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–285, February 1989.