2.5D Visual Speech Synthesis Using Appearance Models

Barry Theobald, Gavin Cawley, John Glauert and Andrew Bangham
School of Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK
{bjt,gcc,jrwg,ab}@sys.uea.ac.uk

Iain Matthews
Robotics Institute, Carnegie Mellon, Pittsburgh, PA 15123, USA
iainm@cs.cmu.edu

Abstract

Two dimensional (2D) shape and appearance models are applied to the problem of creating a near-videorealistic talking head. A speech corpus of a talker uttering a set of phonetically balanced training sentences is analysed using a generative model of the human face. Segments of original parameter trajectories corresponding to the synthesis unit (e.g. triphone), are extracted from a codebook, then normalised, blended, concatenated and smoothed before being applied to the model to give natural, realistic animations of novel utterances. The system provides a 2D image sequence corresponding to the face of a talker. It is also used to animate the face of a 3D avatar by displacing the mesh according to movements of points in the shape model and dynamically texturing the face polygons using the appearance model.

1 Background

Recently, interest in the fields of computer vision and computer graphics have been converging. One particular area that has benefited greatly from this convergence is the realistic animation of human faces. The face is a complex communication device that provides both linguistic and non-linguist cues and we quickly become expert at detecting and recognising subtle changes in the features of the face. As a result videorealistic animation of the face is a very difficult problem.

Traditional graphics-based facial animation systems represent points on the face as vertices in three dimensions and approximate the surface of the face by connecting the vertices. A set of parameters deform the mesh in some controlled manner, where the parameterisation is either direct, as in *terminal analogue* synthesis [12, 13], or indirect, as in *physically-based* synthesis [15, 17]. Graphics-based systems can be efficiently rendered, especially on modern graphics processors, however they tend to lack videorealism. Texture mapping an image of a real face onto the mesh generally is still not enough to convince a viewer that the animated sequence is a real face.

Computer vision and image processing algorithms can improve the videorealism, where animations are driven from images of real faces [6, 9, 11]. Providing the correct lip shape is presented for a given sound and the synthesised movements of the face look natural, these systems are able to achieve a high degree of videorealism. Bregler and co-workers [6] automatically segment existing sequences of a talker into short sequences corresponding to *triphones* and replay these segments in a new order to create novel sequences. Brand [5] and Brooke and Scott [7] use hidden Markov models (HMMs) to learn the characteristics of facial deformations associated with speech production. The trained HMMs are used to generate new sequences, where Brand animates both the speech and expression of a (possibly) novel person, while Brooke and Scott generate image sequences of the mouth region of a single talker. Cosatto and Graf [9] populate a hyper-space of facial examples, where the dimensions of the hyper-space correspond to measurements on a talker's face. Example images are extracted from this compact hyper-space based on the phonetic string to be synthesised and these mouth shape images stitched together with images of other facial regions (eyes, cheeks etc.) to create novel sequences of expressive speech. Ezzat and Poggio report a model-based synthesis technique that creates very realistic speech animation of a talker's mouth [11]. These images of the synthesised mouth movements are composited back into an original video sequence to create believable animated speech.

Image-based animation systems tend to lack the generality of graphics based systems. For example, they are (usually) applied to problem of reanimating the face in existing video sequences. For a full-bodied talking character, graphics methods may be preferable since the face of a complete virtual character (avatar) can be animated while the character performs novel actions, e.g. sign language and other manual gestures.

In this paper we describe an extension to an alternative technique for achieving realistic speech animation based on appearance models, where the system is extended to animate the face of a complete 3D virtual character. A statistical model of the appearance of the face is texture mapped onto a three dimensional mesh model, which in turn is animated by a statistical model of shape. Thus, pose, shape and texture are all animated independently.

2 Data Capture

To ensure the pose of the head remained as constant as possible, the training data was collected using a head mounted camera and transferred from DV tape to computer using an IEEE 1394 compliant capture card with a frame size of 360x288 pixels (one quarter DV-PAL). The audio was captured using the on-camera microphone and digitised at 11025 Hz, 16 bits/sample stereo and was later used to phonetically segment the video using the HTK speech recogniser run in forced-alignment mode [18]. Only a single talker was recorded in a single sitting to remove identity variation and to ensure the lighting was even and constant throughout the entire training video. The speaker held the facial expression as neutral as possible (no emotion) to confine the variation of the facial features to that due to speech. The training data consisted of 279 sentences, comprising approximately twelve minutes of speech data.

3 Modelling the Face

Following the notation of Cootes and co-workers [8] a statistical model of the shape of an object, termed the *point distribution model* (PDM), is trained by manually placing landmarks on a set of images and performing a principal component analysis (PCA) on the coordinates of these landmarks. Typically about 100 points are used for the whole face and 30 images are selected for hand labelling covering a broad range of the mouth shapes associated with speech production. Any training shape can then be approximated using $\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s$, where $\mathbf{x}$ contains the (x,y) coordinate pairs for each landmark, $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{P}_s$ is the matrix of the eigenvectors of the covariance matrix associated with the t_s eigenvalues of the greatest magnitude, chosen to describe some preset percentage of the total variation (typically 95%), and $\mathbf{b}_s$ is a vector of t_s shape parameters.

A statistical model of the appearance of the face is computed by warping the labelled images from the landmarks to the mean shape. This normalises the shape of the face in each image, ensures each example has the same number of pixels and a pixel in one example corresponds to the same feature of the face in all other examples. Typically about 40,000 (RGB) pixels are used. A further PCA is performed on the pixel values within the shape-normalised faces such that any RGB appearance can be approximated using $\mathbf{a} \approx \bar{\mathbf{a}} + \mathbf{P}_a \mathbf{b}_a$, where $\bar{\mathbf{a}}$ is the mean shape-normalised image, $\mathbf{P}_a$ is the matrix of the first t_a eigenvectors of the covariance matrix and $\mathbf{b}_a$ a vector of appearance parameters.

Each image is, therefore, described by a set of shape parameters and a set of appearance parameters, $\mathbf{b}_s$ and $\mathbf{b}_a$ respectively. The shape and appearance spaces are concatenated such that the face in an image maps to a single point in a *face-space*, where some of the dimensions correspond to shape variation and some to appearance variation. We do not project the shape and appearance parameters into a combined model space for synthesis as subjective testing of various appearance models have shown that the most *dynamically* realistic models are comprised of independent shape and appearance models [16].

4 Data Preparation

Given the shape and appearance models, the face in all 34000 video frames is encoded in terms of the parameters b_s and b_a. To project the face onto the principal components requires the landmark positions for each image, which are obtained using the *gradient descent active appearance* search algorithm [2]. This takes as input an image, the shape model and the appearance model, and outputs the corresponding landmarks for each frame. This labelling can be done using any face tracker, however active appearance models and their descendent's have the advantage that they use the same models as used by the synthesiser. Hence the points on the face located by the tracker are exactly the points required by the synthesiser.

Given the landmarks, each image is projected into face-space by computing the shape parameters, warping the image from the landmarks to the mean shape and computing the appearance parameters. Each example image corresponds to a point in face-space and over the course of a sentence the parameters approximate a trajectory through face-space. A continuous parametric representation of this trajectory is obtained using Hermite interpolation [3], and the 279 continuous trajectories, one for each training sentence, are stored in the synthesis codebook. Hermite interpolation is used to fit the data rather than natural cubic splines as the smoothness constraints in the calculation of the natural cubic

spline often results in an over-smoothed fit of the data points. If, say, a point along the parameter trajectory corresponds to mouth opening, overshooting could result in the mouth opening further than actually occurred in the original data and the auditory and visual information could become misaligned.

4.1 Segmenting the Trajectories

The audio component of the training video is passed through the HTK speech recogniser [18], the output of which is a list of the constituent phoneme symbols that form each sentence and their corresponding start and end times. This phonetic information is also stored in the synthesis codebook and is later used to index the parameter trajectories such that segments can be extracted corresponding to individual phonemes.

4.2 Measuring Phoneme Similarity

It is well known that during speech lip shapes depend not only the sound being produced, but also the surrounding sounds — known as phonetic context. The shape and appearance models are used in a sample-based synthesis scheme, so the synthesiser must be able to account for phonemes appearing in unseen contexts. To allow for this a similarity matrix is used to find contexts in the training data that are 'closest' to an unseen context. This similarity matrix is automatically derived from the data and each element contains an objective measure of similarity, in terms of the model parameters, between two given phonemes. This idea is similar to that in [1], except we extend their idea to consider the time variation of the parameters, the degree to which phonemes are modified by context and the relative significance of each model parameter.

To build the matrix, first all observations of each phoneme are gathered and the relevant portions of the original shape and appearance trajectories sampled at five equi-distant points over the duration of the phoneme[1], where the timing information from the speech recogniser is used to index the trajectories. Each observation is then represented as a $(t_s + t_a) \times 5$ matrix of shape and appearance parameters and the mean representation of each phoneme is computed. The distance between any phoneme pair can then be found on a pair-wise basis using,

$$D_{ij} = \sum_k \sum_l \left[\left(v_i P^i_{kl} - v_j P^j_{kl} \right) w_k \right]^2, \tag{1}$$

where D_{ij} is the distance between phonemes i and j. P^i is the mean $(t_s + t_a) \times 5$ matrix representing the i^{th} phoneme and P^j the j^{th} phoneme. The weights v take into account the degree to which the context modifies the lip shape for a phoneme, i.e. how reliable the mean representation is for a phoneme. For each phoneme, its weight is proportional to the total area between the mean trajectory and all observed trajectories. The value w_k is the significance of the k^{th} parameter in the model and is proportional to the variance captured by the corresponding principal component.

Given the matrix of distance values, the similarities are computed using

$$S_{ij} = e^{-\gamma D_{ij}}. \tag{2}$$

[1] The choice of sampling at 5 equi-distant points follows [1].

The range of similarity is 0 (maximally dissimilar), to 1 (identical) and the variable γ controls the spread of similarity values over the range (0,1). This similarity matrix is stored with the parameter trajectories and phoneme timing information in the synthesis codebook. Some typical similarity values are shown in Table 1.

Phoneme	Rank 1		Rank 2		Rank 3	
m	p	0.869	b	0.850	w	0.830
f	v	0.808	s	0.621	dʒ	0.619
t	d	0.967	ɪ	0.900	z	0.894
tʃ	dʒ	0.898	ʃ	0.852	s	0.767

Table 1: Some typical phoneme similarity scores. The column Rank 1 is the most similar phoneme with its similarity score, Rank 2 the second most similar and so on. Generally the most similar phonemes belong to the same class of sound, for example the bilabials /b/, /m/ and /p/ are all considered similar, as with the labio-dental fricatives, /f/ and /v/.

5 Synthesis

A visual sequence corresponding to a new utterance is synthesised by first converting a text stream to a list of phonemes and durations. This can either be from analysis of a real (unseen) utterance, or derived from a text-to-speech (TTS) synthesiser. For each phoneme to be synthesised, the original training data is searched for the *n* examples of that phoneme in the most similar contexts found in the codebook using

$$\mathbf{s}_j = \sum_{i=1}^{C} \frac{\mathbf{S}_{l_i j}}{i+1} + \sum_{i=1}^{C} \frac{\mathbf{S}_{r_i j}}{i+1}, \tag{3}$$

where $\mathbf{s}_j$ is the similarity between the desired context and the j^{th} context in the codebook, C is the context width, $\mathbf{S}_{l_i j}$ is the similarity between the i^{th} left phoneme in the j^{th} codebook context and the corresponding phoneme in the desired context, $\mathbf{S}_{r_i j}$ is the similarity between the i^{th} right phoneme of the j^{th} codebook context and the corresponding phoneme in the desired context. This similarity score is attractive since it allows the context width to be easily varied by simply changing an input parameter to the synthesiser (C), the structure of the synthesiser itself requires no modification. In the results presented here a context width of $C = 1$ is used, hence, the synthesis unit is the triphone. Given the *n* closest matches in the codebook for each synthesis phoneme, the corresponding portions of the original parameter trajectories are extracted and temporally warped to the desired duration. A weighted average of these normalised trajectories is computed to give a new trajectory in face-space, where the weights are proportional to the similarity of the codebook context to the synthesis context, ensuring the most similar contexts receive more weight.

The new phoneme trajectories in face-space are concatenated to form a trajectory for the entire sentence, which is sampled at the original frame rate. Since no smoothness constraints were imposed on examples selected from the codebook, smoothing splines [10] are fitted through the model parameters to ensure a smooth transition between synthesis units and the smoothed parameters are applied to the model to produce the synthetic

image sequence of the talking face. The synthesiser itself outputs a sequence of 2D landmarks and a sequence of shape-normalised images. The final synthesised image frames are created by warping the shape-normalised images to the corresponding landmarks.

Example parameter trajectories are shown in Figure 1, where the trajectory for the first parameter for the shape and appearance models are shown for an original (novel) sequence and the synthesised equivalent. While there are systematic differences between the trajectories, the overall shape is generally correct. Formal subjective testing is required in order to determine the significance of the differences between these trajectories. However, informal testing suggests the differences are not critical. Formal subjective tests are currently underway and results will be presented in a separate paper. A comparison of original and synthesised faces from a real sequence and the corresponding synthesiser output are shown in Figure 2. The data for the original sequence was not included in the synthesis codebook.

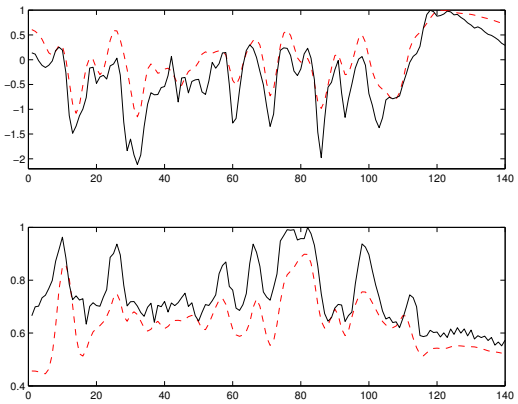

Figure 1: Upper plot shows the first shape model parameter trajectory from an original sequence (solid curve) and a synthetic sequence (dashed curve). The lower plot shows the same information, but for the first appearance parameter. The trajectories correspond to the phrase “Charlie brought his dog out but their only pure intent was to catch churchgoers wearing turquoise.”

The synthesis method described here for creating near-videorealistic synthetic visual speech sequences has the advantage over traditional image-based systems in that the manipulation of the original data is much easier in terms of the model parameters than the original images. The resultant sequences are still only 2D image sequences of a talking face however. It just happens that the images are created by the generative model rather than obtained directly from a camera. The next section describes how the synthesiser can be easily extended to animate a 3D mesh model, providing near-videorealistic 2.5D animations.

5.1 Extending the Synthesis to 2.5D

The synthesiser described in Section 5 provides very realistic 2D speech animation of the human face. The resultant synthetic faces can be composited back into an original sequence, as with other 2D synthesis systems [6, 11]. However, it is desirable to animate

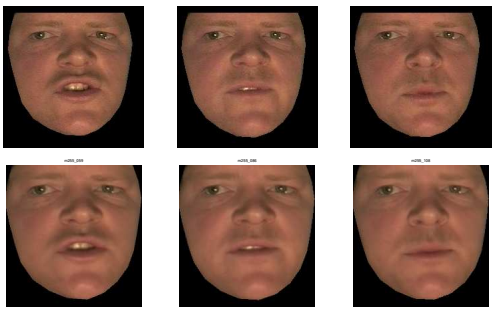

Figure 2: The top rows shows pixel values extracted from selected video frames from a real video sequence not used in training, while the bottom row shows the corresponding face output by the synthesiser.

the face of a full-bodied 3D virtual character, rather than simply re-animating the mouth in an existing sequence.

Blanz and Vetter [4] describe how a database of Cyberware scans can be adapted to a single image of an individual to provide a 3D photorealistic representation of the face. Instead we adopt a technique based on scattered data interpolation used in [14] for animating facial expression. The 3D coordinates of a sparse set of points defined on the face of an individual are recovered from multiple camera views. These sparse points are then used to adapt a dense generic 3D mesh to the individual. Here, we use the same model used by the face tracker and synthesiser, where the sparse shape model points are used to drive the dense 3D mesh and the shape-normalised image provides a *dynamic texture map*. These texture maps are warped to the 3D vertices of the face mesh rather than the 2D points of the shape model, providing near-videorealism in three dimensions. The actual animations produced by the synthesiser are essentially 2.5D since the shape model contains no depth variation. The resultant animations are however still very realistic for moderate rotations of the head as the depth cues are captured in the subtle changes of the dynamic texture map.

First, a correspondence must be defined between the N 2D landmarks in the shape model and the $M >> N$ vertices of the 3D mesh. This is done manually prior to synthesis and informs the synthesiser which vertex belongs to which point in the shape model. Vertices on the 3D mesh mapped to a point in the shape model are known as *constrained* vertices and the displacements for these vertices are known, they take the coordinates of the corresponding shape model points. The displacements for the constrained vertices are given by:

$$\mathbf{u}_i = \mathbf{p}_i - \mathbf{p}_i^{(0)}, \tag{4}$$

where $\mathbf{p}^{(0)}$ is the 3D mesh in the default position, i.e. adapted to the mean shape in the shape model, and $\mathbf{p}_i$ are the new 3D coordinates for the i^{th} constrained vertex. A smooth vector-valued function that fits the known displacements, $f(\mathbf{p}_i) = \mathbf{u}_i$ is defined such that the displacements of the remaining, unconstrained vertices can be found using $f(\mathbf{p}_j) = \mathbf{u}_j$.

A radially symmetric basis function is used in [14], which falls off smoothly with distance, thus the displacement of unconstrained vertices are more influenced by the dis-

placement of constrained vertices lying closer by. The function $f(p)$ is defined as

$$f(p) = \sum_i \mathbf{c}_i \phi(\| \mathbf{p} - \mathbf{p}_i \|), \tag{5}$$

where the basis function takes the form $\phi(r) = e^{-r/64}$ and the coefficients $\mathbf{c}$ are found by multiplying the (x,y,z) coordinates of $\mathbf{u}_i$ with the matrix Φ^{-1}, where $\Phi_{ik} = \phi(\| \mathbf{p}_i - \mathbf{p}_k \|)$, with $\mathbf{p}_i$ the i^{th} constrained vertex and $\mathbf{p}_k$ the k^{th} constrained vertex.

The original synthesiser training data was captured using a head mounted camera to minimise unwanted pose variation from the face model. In the synthetic sequences, pose information (translation and rotation) can be applied to the 3D mesh prior to rendering the face, hence the pose of the face is independent of the synthesis parameters. The mesh used to drive the model need not contain only a face, it could form part of a full virtual character, shown in Figure 4. In this instance vertices are tagged prior to synthesis as belonging to the face of the avatar, or not. Those not forming the face are ignored, while those belonging to the face are displaced as described above. Example frames from an animated sequence using a generic mesh model are shown in Figure 3 and an example of a full bodied talking avatar is shown in Figure 4.

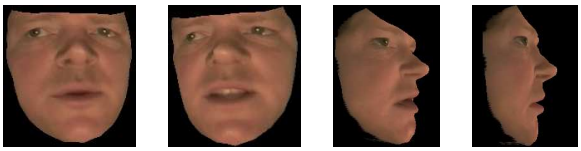

Figure 3: Example frames from a sequence, where a generic mesh is deformed according to the 2D shape model points and textured the with shape-free appearance images output by the synthesiser.

Figure 4: Example frames from a sequence, where the face of a complete avatar is animated using an appearance model. Here, the face model is clearly delineated by not matching the mesh textures to the face. Notice (left and right) that the teeth and other subtle changes have been captured.

6 Future Work

Future work will include an investigation of how expressive speech can be animated using the model. Currently the synthesiser is trained on speech without emotional context. One

approach to animating expressive speech would be to include existing graphics rules, for example Waters' muscle model [17], and apply the graphics rules to the 3D mesh after the speech animation has been generated. A second approach could be to capture a database of images with emotional expression in the same session as a speech corpus is captured. A separate shape and appearance model could then be trained on this database, and the leading modes of variation added to the speech model. One of the major limitations of image-based synthesis is the lack of generalisation - only the face(s) in the synthesiser corpus may be animated. Since our animation parameters are offsets from the mean shape and appearance, we will investigate how displacing the mean to a new position in face-space affects the perceptual quality of the synthesiser output for a novel face.

7 Conclusions

In this paper we have presented an alternative to existing techniques for creating highly realistic synthetic visual speech. The synthesiser generates a new trajectory in face-space corresponding to a novel utterance from example parameter trajectories in a corpus. The parameters are applied to the model to create a 2D set of landmarks and a shape-normalised image. The final synthetic video frame is generated by warping the shape-normalised image to the 2D landmarks, or by adapting a generic 3D mesh to the landmarks and warping the shape-normalised image to the new mesh vertices. We have conducted formal subjective tests of the naturalness of the synthesiser output, which will be published in a separate paper. Demos of the system can be found on our web-page at http://www.sys.uea.ac.uk/˜bjt.

8 Acknowledgements

The authors are grateful to Vince Jennings for his help in texturing the appearance model onto the face of the avatar, Figure 4.

References

[1] L.M. Arslan and D. Talkin. 3D face point trajectory synthesis using an automatically derived visual phoneme similarity matrix. In *Proceedings of Auditory-Visual Speech Processing*, pages 175–180, Terrigal, Australia, December 1998.

[2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, Kauai, Hawaii, 2001.

[3] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of SIGGRAPH*, pages 187–194, Los Angeles, California, August 1999.

[5] M. Brand. Voice puppetry. In *Proceedings of SIGGRAPH*, pages 21–28, Los Angeles, California, 1999.

[6] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Computer Graphics Annual Conference Series (SIGGRAPH)*, pages 353–360, Los Angeles, California, August 1997.

[7] N.M. Brooke and S.D. Scott. Two- and three-dimensional audio-visual speech synthesis. In *Proceedings of Auditory-Visual Speech Processing*, pages 213–218, Terrigal, Australia, December 1998.

[8] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, *Proceedings of the European Conference on Computer Vision*, volume 2, pages 484–498, Freiburg, Germany, 1998. Springer-Verlag.

[9] E. Cosatto and H.P. Graf. Sample-based synthesis of photorealistic talking heads. In *Proceedings of Computer Animation*, pages 103–110, Philadelphia, Pennsylvania, June 1998.

[10] C. de Boor. Calculation of the smoothing spline with weighted roughness measure. *Mathematical Models and Methods in Applied Sciences*, 11(1):33–41, 2001.

[11] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. In *Proceedings of SIGGRAPH*, pages 388–398, San Antonio, Texas, July 2002.

[12] D. Massaro. *Perceiving Talking Faces*. The MIT Press, 1998.

[13] F.I. Parke and K. Waters. *Comptuer Facial Animation*. A K Peters, 1996.

[14] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of SIGGRAPH*, pages 75–84, Orlando, Florida, 1998.

[15] S. Platt and N. Badler. Animating facial expression. *Computer Graphics*, 15(3):245–252, 1981.

[16] B.J. Theobald, G.C. Cawley, I.A. Matthews, and J.A. Bangham. Near-videorealistic synthetic visual speech using non-rigid appearance models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 800–803, Hong Kong, 2003.

[17] K. Waters. A muscle model for animating three-dimensional facial expressions. *Proceedings of SIGGRAPH*, 21(4):17–24, 1987.

[18] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book*. Entropic Ltd., Cambridge, 1999.