Trainable Videorealistic Speech Animation

Tony Ezzat Gadi Geiger Tomaso Poggio
Center for Biological and Computational Learning
Massachusetts Institute of Technology
Cambridge, MA
tonebone, gadi, tp@ai.mit.edu

Abstract

We describe how to create with machine learning techniques a generative, videorealistic, speech animation module. A human subject is first recorded using a videocamera as he/she utters a pre-determined speech corpus. After processing the corpus automatically, a visual speech module is learned from the data that is capable of synthesizing the human subject's mouth uttering entirely novel utterances that were not recorded in the original video. The synthesized utterance is re-composited onto a background sequence which contains natural head and eye movement. The final output is videorealistic in the sense that it looks like a video camera recording of the subject. At run time, the input to the system can be either real audio sequences or synthetic audio produced by a text-to-speech system, as long as they have been phonetically aligned.

1. Overview

Is it possible to record a human subject with a video camera, process the recorded data automatically, and then reanimate that subject uttering entirely novel utterances which were not included in the original corpus? In this work, we present such a technique for achieving videorealistic speech animation. [1].

We choose to focus our efforts in this work on the issues related to the synthesis of novel video, and not on novel audio synthesis. Thus, novel audio needs to be provided as input to our system. This audio can be either real human audio (from the same subject or a different subject), or synthetic audio produced by a text-to-speech system. All that is required by our system is that the audio be phonetically transcribed and aligned.In the case of synthetic audio from TTS systems, this phonetic alignment is readily available from the TTS system itself [6]. In the case of real audio, publicly available phonetic alignment systems [22] may be used.

1 A longer version of this paper appeared in [16]

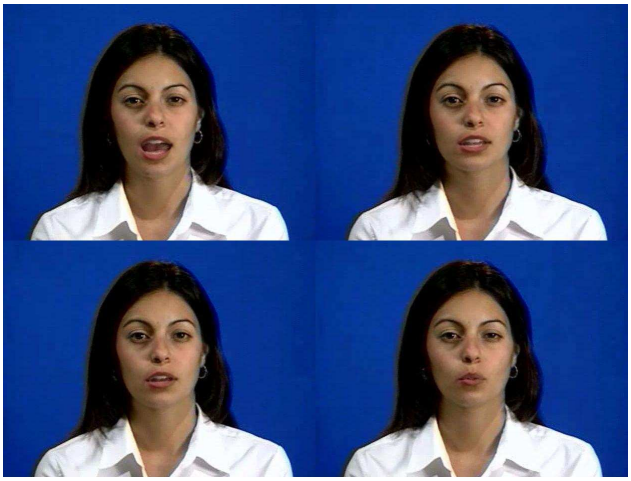

Figure 1. Some of the synthetic facial configurations output by our system.

Our visual speech processing system is composed of two modules: The first module is the *multidimensional morphable model* (MMM), which is capable of morphing between a small set of prototype mouth images to synthesize new, previously unseen mouth configurations. The second component is a *trajectory synthesis* module, which uses regularization [19] [36] to synthesize smooth trajectories in MMM space for any specified utterance. The parameters of the trajectory synthesis module are trained automatically from the recorded corpus using gradient descent learning.

Application scenarios for videorealistic speech animation include: user-interface agents for desktops, TVs, or cell-phones; digital actors in movies; virtual avatars in chatrooms; very low bitrate coding schemes (such as MPEG4); and studies of visual speech production and perception. The recorded subjects can be regular people, celebrities, ex-presidents, or infamous terrorists.

In the following section, we begin by first reviewing the

relevant prior work and motivating our approach.

2. Background

2.1. Facial Modeling and Speech Animation

One approach to model facial geometry is to use *3D methods*. Parke [28] was one of the earliest to adopt such an approach by creating a polygonal facial model. To increase the visual realism of the underlying facial model, the facial geometry is frequently scanned in using Cyberware laser scanners. Additionally, a texture-map of the face extracted by the Cyberware scanner may be mapped onto the three-dimensional geometry [25]. Guenter [20] demonstrated recent attempts at obtaining 3D face geometry from multiple photographs using photogrammetric techniques. Pighin et al. [30] captured face geometry and textures by fitting a generic face model to a number of photographs. Blanz and Vetter [8] demonstrated how a large database of Cyberware scans may be morphed to obtain face geometry from a single photograph.

An alternative to the 3D modeling approach is to model the talking face using *image-based* techniques, where the talking facial model is constructed using a collection of example images captured of the human subject. Bregler, Covell, and Slaney [10] describe an image-based facial animation system called Video Rewrite in which the recorded video is broken into a set of smaller audiovisual basis units. Each one of these short sequences is a *triphone* segment, and a large database with all the acquired triphones is built. A new audiovisual sentence is constructed by *concatenating* the appropriate triphone sequences from the database together.

The approach used in this work presents another approach to solving the video synthesis problem which has the capacity to *generate novel video from a small number of examples* as well as the capacity *to model how the mouth moves*. This approach is based on the use of a *multidimensional morphable model* (MMM), which is capable of multdimensional morphing between various lip images to synthesize new, previously unseen lip configurations. MMM's have already been introduced in other works [31] [3] [13] [23] [24] [8] [7]. In this work, we develop an MMM variant and show its utility for facial animation.

In terms of speech animation, techniques have traditionally included both keyframing methods [28] [29] [12] [26] and physics-based methods [37] [25], and have been extended more recently to include machine learning methods [9] [27] [11].

In this work, we present a *trajectory synthesis* module to address the issues of synthesizing mouth trajectories with correct motion, smoothness, dynamics, and coarticulation effects. This module maps from an input stream of phonemes (with their respective frame durations) to a *trajectory* of MMM shape-appearance parameters. This trajectory is then fed into the MMM to synthesize the final visual stream that represents the talking face.

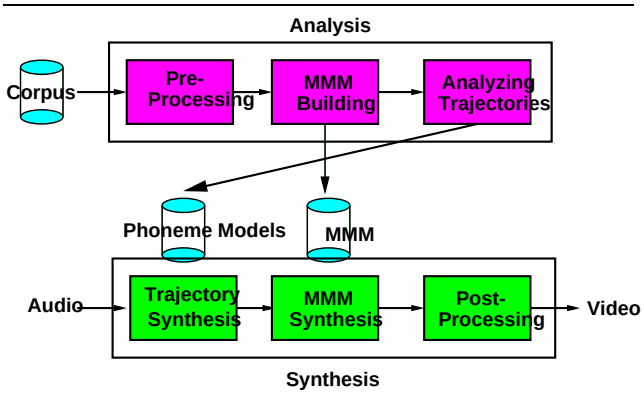

Figure 2. An overview of our videorealistic speech animation system.

3. System Overview

An overview of our system is shown in Figure 2. After recording the corpus (Section 4), *analysis* is performed to produce the final visual speech module. Analysis itself consists of three sub-steps: First, the corpus is pre-processed (Section 5) to align the audio and normalize the images to remove head movement. Next, the MMM is created from the images in the corpus (Section 6.2). Finally, the corpus sequences are analyzed to produce the phonetic models used by the trajectory synthesis module (Sections 6.4 and 7.2).

Given a novel audio stream that is phonetically aligned, synthesis proceeds in three steps: First, the trajectory synthesis module is used to synthesize the trajectory in MMM space using the trained phonetic models (Section 7). Secondly, the MMM is used to synthesize the novel visual stream from the trajectory parameters (Section 6.3). Finally, the post-processing stage composites the novel mouth movement onto a background sequence containing natural eye and head movements (Section 8).

4. Corpus

An audiovisual corpus of a human subject uttering various utterances was recorded. Recording was performed at a TV studio against a blue "chroma-key" background with a standard Sony analog TV camera. The data was subsequently digitized at a 29.97 fps NTSC frame rate with an image resolution of 640 by 480 and an audio resolution of 44.1KHz. The final sequences were stored as Quicktime sequences compressed using a Sorenson coder. The recorded corpus lasts for 15 minutes, and is composed of approxi-

mately 30000 frames. The recorded corpus consisted of 152 1-syllable and 156 2-syllable words, In addition, the corpus included 105 short sentences.

5. Pre-Processing

The recorded corpus data needs to be pre-processed in several ways before it may be processed effectively for re-animation.

Firstly, the audio needs to be phonetically aligned in order to be able to associate a phoneme for each image in the corpus. We perform audio alignment on all the recorded sequences using the CMU Sphinx system [22], which is publicly available.

Secondly, each image in the corpus needs to be *normalized* to remove any head movement. Since the head motion is small, we make the simplifying assumption that it can be approximated as the perspective motion of a plane lying on the surface of the face, and remove it by perspective warping the current frame with respect to a reference frame [16].

6. Multidimensional Morphable Models

6.1. Definition

An MMM consists of a *set of prototype images* $\{I_i\}_{i=1}^N$ that represent the various lip textures that will be encapsulated by the MMM. One image is designated arbitrarily to be the *reference image* I_1.

Additionally, the MMM consists of a set of *prototype flows* $\{C_i\}_{i=1}^N$ that represent the correspondences between the reference image I_1 and the other prototype images in the MMM. The correspondence from the reference image to itself, C_1, is designated to be an empty, zero, flow.

In this work, we choose to represent the correspondence maps using relative displacement vectors:

$$C_i(\mathbf{p}) = \{d_x^i(\mathbf{p}), d_y^i(\mathbf{p})\}. \quad (1)$$

A pixel in image I_1 at position $\mathbf{p} = (x, y)$ corresponds to a pixel in image I_i at position $(x + d_x^i(x, y), y + d_y^i(x, y))$.

In this work, we make use of *optical flow* [21] [1] [2] algorithms to estimate this motion. This motion is captured as a two-dimensional array of displacement vectors, in the same exact format shown in Equation 1.

6.2. Building an MMM

An MMM must be constructed automatically from a recorded corpus of $\{I_j\}_{j=1}^S$ images. The two main tasks involved are to choose the image prototypes $\{I_i\}_{i=1}^N$, and to compute the correspondence $\{C_i\}_{i=1}^N$ between them. We discuss the steps to do this briefly below. Note that the following operations are performed on the entire face region, although they need only be performed on the region around the mouth.

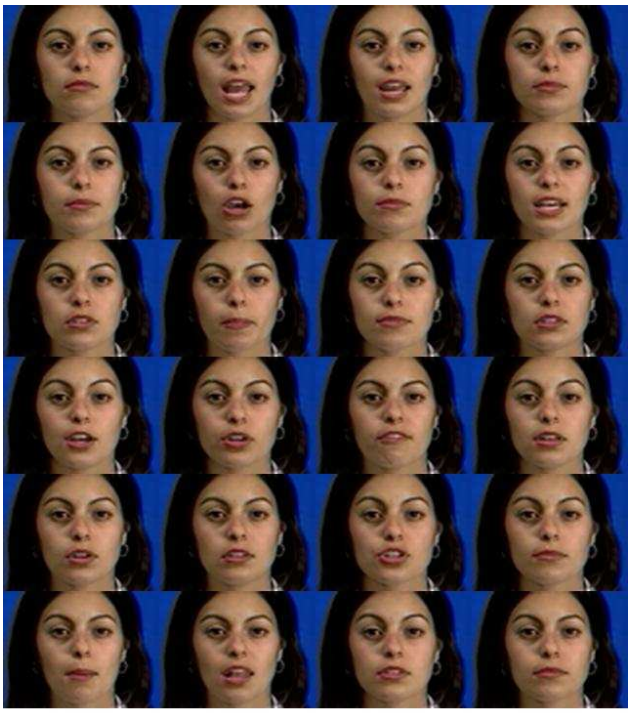

Figure 3. 24 of the 46 image prototypes included in the MMM. The reference image is the top left frame.

6.2.1. PCA For the purpose of more efficient processing, principal component analysis (PCA) is first performed on all the images of the recorded video corpus. PCA allows each image in the video corpus to be represented using a set of low-dimensional parameters. This set of low-dimensional parameters may thus be easily loaded into memory and processed efficiently in the subsequent clustering and Dijkstra steps. We adopt an on-line PCA method, termed EM-PCA [32] which allows us to perform PCA on the images in the corpus without loading them all into memory.

Performing EM-PCA produces a set of D 624x472 principal components and a matrix Σ of eigenvalues. In this work, $D = 15$ PCA bases are retained. The images in the video corpus are subsequently projected on the principal components, and each image I_j is represented with a D-dimensional parameter vector p_j.

6.2.2. K-means Clustering Selection of the prototype images is performed using *k-means clustering* [5]. The algorithm is applied directly on the $\{p_j\}_{j=1}^S$ low dimensional PCA parameters, producing N cluster centers. Typically the cluster centers extracted by k-means clustering do not coincide with actual image datapoints, so the nearest images in the dataset to the computed clus-

ter centers are chosen to be the final image prototypes $\{I_i\}_{i=1}^N$ for use in our MMM. The distance metric used between two points p_m and p_n is the *Mahalanobis distance metric*:

$$d(p_m, p_n) = (p_m - p_n)^T \Sigma^{-1} (p_m - p_n) \qquad (2)$$

where Σ is the afore-mentioned matrix of eigenvalues extracted by the EM-PCA procedure.

We selected $N = 46$ image prototypes in this work, which are partly shown in Figure 3. The top left image is the reference image I_1. There is nothing magical about our choice of 46 prototypes, which is in keeping with the typical number of visemes other researchers have used [33] [18]. It should be noted, however, that the 46 prototypes have no explicit relationship to visemes, and instead form a simple *basis set* of image textures.

6.2.3. Dijkstra After the $N = 46$ image prototypes are chosen, the next step in building an MMM is to compute correspondence between the reference image I_1 and all the other prototypes. Although it is in principle possible to compute *direct* optical flow between the images, we have found that direct application of optical flow is not capable of estimating good correspondence when the underlying lip displacements between images are greater than 5 pixels.

To compute good correspondence between prototypes, we construct the *corpus graph* representation of the corpus: A corpus graph is an S-by-S sparse adjacency graph matrix in which each frame in the corpus is represented as a node in a graph connected to k nearest images. The k nearest images are chosen using the *k-nearest neighbors* algorithm [5], and the distance metric used is the Mahalanobis distance in Equation 2 applied to the PCA parameters *p*. We set $k = 20$ in this work.

After the corpus graph is computed, the *Dijkstra* shortest path algorithm [14] [35] is used to compute the shortest path between the reference example I_1 and the other chosen image prototypes I_i. Each shortest path produced by the Dijkstra algorithm is a list of images from the corpus that cumulatively represent the shortest deformation path from I_1 to I_i as measured by the Mahalanobis distance. Concatenated optical flow from I_1 to I_i is then computed along the intermediate images produced by the Dijkstra algorithm (see [16] for details on concatenated optical flow). Since there are 46 images, $N = 46$ correspondences $\{C_i\}_{i=1}^N$ are computed in this fashion from the reference image I_1 to the other image prototypes $\{I_i\}_{i=1}^N$.

6.3. Synthesis

The goal of synthesis is to map from the multidimensional parameter space (α, β) to an image which lies at that position in MMM space. Since there are 46 correspondences, α is a 46-dimensional parameter vector that controls mouth shape. Similarly, since there are 46 image prototypes, β is a 46-dimensional parameter vector that controls mouth texture. The total dimensionality of (α, β) is 92.

Figure 4. Top: Original images from our corpus. Bottom: Corresponding synthetic images generated by our system.

Synthesis first proceeds by synthesizing a new correspondence C^{synth} using *linear combination* of the prototype flows C_i:

$$C_1^{synth} = \sum_{i=1}^{N} \alpha_i C_i. \qquad (3)$$

The subscript 1 in Equation 3 above is used to emphasize that C_1^{synth} originates from the reference image I_1, since all the prototype flows are taken with I_1 as reference.

Forward warping may be used to push the pixels of the reference image I_1 along the synthesized correspondence vector C_1^{synth}. Notationally, we denote the forward warping operation as an operator $\mathbf{W}(I, C)$ that operates on an image I and a correspondence map C (see Appendix B in [16] for details on forward warping).

However, a single forward warp will not utilize the image texture from *all* the examples. In order to take into account all image texture, a *correspondence re-orientation* procedure first described in [4] is adopted that re-orients the synthesized correspondence vector C_1^{synth} so that it originates from each of the other example images I_i:

$$C_i^{synth} = \mathbf{W}(C_1^{synth} - C_i, C_i). \qquad (4)$$

Re-orientation is performed for all examples in the example set.

The third step in synthesis is to warp the prototype images I_i along the re-oriented flows C_i^{synth} to generate a set of N warped image textures I_i^{warped}:

$$I_i^{warped} = \mathbf{W}(I_i, C_i^{synth}). \qquad (5)$$

The fourth and final step is to blend the warped images I_i^{warped} using the β parameters to yield the final morphed image:

$$I^{morph} = \sum_{i=1}^{N} \beta_i I_i^{warped}. \quad (6)$$

Combining Equations 3 through 6 together, our MMM synthesis may be written as follows:

$$I^{morph}(\alpha, \beta) = \sum_{i=1}^{N} \beta_i \mathbf{W}(I_i, \mathbf{W}(\sum_{j=1}^{N} \alpha_j C_j - C_i, C_i)). \quad (7)$$

Empirically we have found that the MMM synthesis technique is capable of surprisingly realistic re-synthesis of lips, teeth, and tongue. However, the blending of multiple images in the MMM for synthesis tends to blur out some of the finer details in the teeth and tongue (See Appendix C in [16] for a discussion of synthesis blur). Shown in Figure 4 are some of the synthetic images produced by our system, along with their real counterparts for comparison.

6.4. Analysis

The goal of analysis is to *project* the entire recorded corpus $\{I_j\}_{j=1}^{S}$ onto the constructed MMM, and produce a time series of $(\alpha_j, \beta_j)_{j=1}^{S}$ parameters that represent trajectories of the original mouth motion in MMM space.

In addition to the image I^{novel} to be analyzed, our analysis method requires that the correspondence C^{novel} from the reference image I_1 in the MMM to the novel image I^{novel} be computed beforehand. In our case, most of the novel imagery to be analyzed will be from the recorded video corpus itself, so we employ the Dijkstra approach discussed in Section 6.2.3 to compute good quality correspondences between the reference image I_1 and I_{novel}.

Given a novel image I^{novel} and its associated correspondence C^{novel}, the first step of the analysis algorithm is to estimate the parameters α which minimize

$$\|C^{novel} - \sum_{i=1}^{N} \alpha_i C_i\|. \quad (8)$$

This is solved using the pseudo-inverse:

$$\alpha = (C^T C)^{-1} C^T C^{novel} \quad (9)$$

where C above is a matrix containing all the prototype correspondences $\{C_i\}_{i=1}^{N}$.

After the parameters α are estimated, N image warps are synthesized in the same manner as described in Section 6.3 using flow-reorientation and warping:

$$I_i^{warp} = \mathbf{W}(I_i, \mathbf{W}(\sum_{i=1}^{N} \alpha_i C_i - C_i, C_i)). \quad (10)$$

The final step in analysis is to estimate the values of β as the values which minimize

$$\|I^{novel} - \sum_{i=1}^{N} \beta_i I_i^{warp}\| \quad \texttt{subject to} \quad \beta_i > 0 \ \forall i \ \texttt{and} \ \sum_{i=1}^{N} \beta_i = 1. \quad (11)$$

The non-negativity constraint above on the β_i parameters ensures that pixel values are not negated. The normalization constraint ensures that the β_i parameters are computed in a normalized manner for each frame, which prevents brightness flickering during synthesis. Equation 11, which involves the minimization of a quadratic cost function subject to constraints, is solved using quadratic programming methods. In this work, we use the Matlab function `quadprog`.

Each utterance in the corpus is analyzed with respect to the 92-dimensional MMM created in Section 6.2, yielding a set of $z_t = (\alpha_t, \beta_t)$ parameters for each utterance. Analysis takes on the order of 15 seconds per frame on a circa 1998 450 MHz Pentium II machine. Shown in Figure 5 in solid blue are example analyzed trajectories for α_{12} and β_{28} computed for the word `tabloid`.

7. Trajectory Synthesis

7.1. Overview

The goal of trajectory synthesis is to map from an input *phone stream* $\{P_t\}$ to a *trajectory* $y_t = (\alpha_t, \beta_t)$ of parameters in MMM space. After the parameters are synthesized, Equation 7 from Section 6.3 is used to create the final visual stream that represents the talking face.

The phone stream is a stream of phonemes $\{P_t\}$ representing that phonetic transcription of the utterance. For example, the word `one` may be represented by a phone stream $\{P_t\}_{t=1}^{15}$ = `(\w\, \w\, \w\, \w\, \uh\, \uh\, \uh\, \uh\, \uh\, \uh\, \n\, \n\, \n\, \n\, \n)`. Each element in the phone stream represents one image frame. We define T to be the length of the entire utterance in frames.

Since the audio is aligned, it is possible to examine all the flow and texture parameters for any particular phoneme. Evaluation of the analyzed parameters from the corpus reveals that parameters representing the same phoneme tend to *cluster* in MMM space. We represent each phoneme p mathematically as a multidimensional Gaussian with mean μ_p and diagonal covariance Σ_p. Separate means and covariances are estimated for the flow and texture parameters.

The trajectory synthesis problem is framed mathematically as a *regularization* problem [19] [36]. The goal is to synthesize a trajectory y which minimizes an objective function E consisting of a *target term* and a *smoothness term*:

$$E = \underbrace{(y-\mu)^T D^T \Sigma^{-1} D(y-\mu)}_{target\ term} + \lambda \underbrace{y^T W^T W y}_{smoothness}. \quad (12)$$

The desired trajectory y is a vertical concatenation of the individual $y_t = \alpha_t$ terms at each time step (or $y_t = \beta_t$, since we treat flow and texture parameters separately):

$$y = \begin{bmatrix} y_t \\ \vdots \\ y_T \end{bmatrix} \quad (13)$$

The target term consists of the relevant means μ and covariances Σ constructed from the phone stream:

$$\mu = \begin{bmatrix} \mu_{P_t} \\ \vdots \\ \mu_{P_T} \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{P_t} & & \\ & \ddots & \\ & & \Sigma_{P_T} \end{bmatrix} \quad (14)$$

The matrix D is a duration-weighting matrix which emphasizes the shorter phonemes and de-emphasizes the longer ones, so that the objective function is not heavily skewed by the phonemes of longer duration:

$$D = \begin{bmatrix} \sqrt{I - \frac{D_{P_1}}{T}} & & & \\ & \sqrt{I - \frac{D_{P_2}}{T}} & & \\ & & \ddots & \\ & & & \sqrt{I - \frac{D_{P_T}}{T}} \end{bmatrix} \quad (15)$$

One possible smoothness term consists of the first order difference operator:

$$W = \begin{bmatrix} -I & I & & & \\ & -I & I & & \\ & & \ddots & & \\ & & & -I & I \end{bmatrix} \quad (16)$$

Higher orders of smoothness are formed by repeatedly multiplying W with itself: second order $W^T W^T W W$, third order $W^T W^T W^T W W W$, and so on.

Finally, the regularizer λ determines the trade-off between both terms.

Taking the derivative of Equation 12 and minimizing yields the following equation for synthesis:

$$(D^T \Sigma^{-1} D + \lambda W^T W) y = D^T \Sigma^{-1} D \mu. \quad (17)$$

Given known means μ, covariances Σ, and regularizer λ, synthesis is simply a matter of plugging them into Equation 17 and solving for y using Gaussian elimination. This is done separately for the flow and the texture parameters. In our experiments a regularizer of degree four yielding *multivariate additive septic splines* [36] gave satisfactory results (see next subsection).

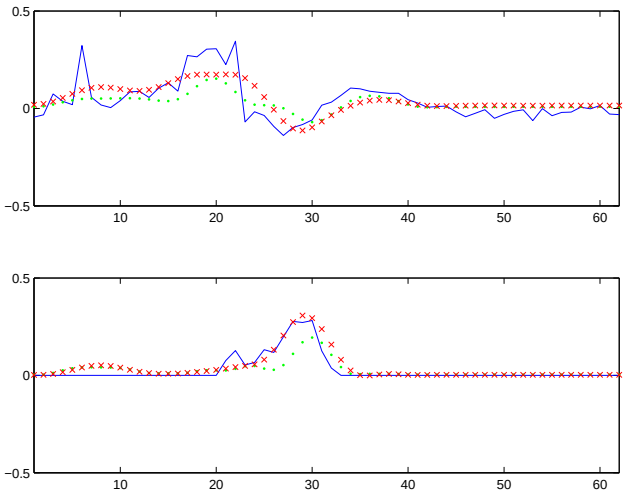

Figure 5. Top: The analyzed trajectory for α_{12} (in solid blue), compared with the synthesized trajectory for α_{12} before training (in green dots) and after training (in red crosses). Bottom: Same as above, but the trajectory is for β_{28}. Both trajectories are from the word `tabloid`.

7.2. Training

The means μ_p and covariances Σ_p for each phone p are initialized directly from the data using sample means and covariances. However, the sample estimates tend to average out the mouth movement so that it looks under-articulated. As a consequence, there is a need to adjust the means and variances to better reflect the training data.

Gradient descent learning [5] is employed to adjust the mean and covariances. First, the Euclidean error metric is chosen to represent the error between the original utterance z and the synthetic utterance y:

$$E = (z-y)^T (z-y). \quad (18)$$

The parameters $\{\mu_p, \Sigma_p\}$ need to be changed to minimize this objective function E. The chain rule may be used to derive the relationship between E and the parameters:

$$\frac{\partial E}{\partial \mu_i} = \left(\frac{\partial E}{\partial y}\right)^T \left(\frac{\partial y}{\partial \mu_i}\right) \quad (19)$$

$$\frac{\partial E}{\partial \sigma_{ij}} = \left(\frac{\partial E}{\partial y}\right)^T \left(\frac{\partial y}{\partial \sigma_{ij}}\right). \quad (20)$$

Gradient descent is performed by changing the previous values of the parameters according to the computed gradi-

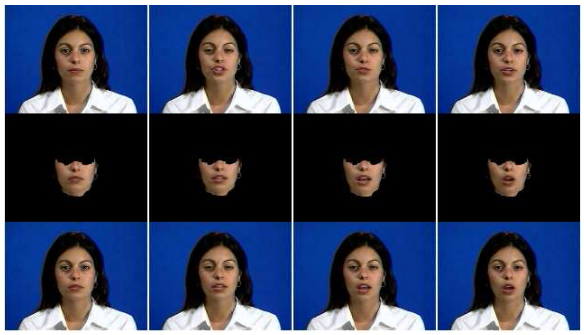

Figure 6. The background compositing process: Top: A background sequence with natural head and eye movement. Middle: A sequence generated from our system, with the desired mouth movement and appropriate masking. Bottom: The final composited sequence with the desired mouth movement, but with the natural head and eye movements of the background sequence. Head and eye masks are used to guide the compositing process.

ent:

$$\mu^{new} = \mu^{old} - \eta \frac{\partial E}{\partial \mu} \quad (21)$$

$$\Sigma^{new} = \Sigma^{old} - \eta \frac{\partial E}{\partial \Sigma}. \quad (22)$$

Cross-validation sessions were performed to evaluate the appropriate value of λ and the correct level of smoothness W to use. The learning rate η was set to 0.00001 for all trials, and 10 iterations performed. The results showed that the optimal smoothness operator is *fourth order* and the optimal regularizer is $\lambda = 1000$. Figure 5 depicts synthesized trajectories for the α_{12} and β_{28} parameters before training (in green dots) and after training (in red crosses) for these optimal values of W and λ.

8. Post-Processing

Due to the head and eye normalization that was performed during the pre-processing stage, the final animations generated by our system exhibit movement only in the mouth region. This leads to an unnerving "zombie"-like quality to the final animations. As in [15] [10], we address this issue by compositing the synthesized mouth onto a background sequence which contains natural head and eye movement.

9. Computational Issues

To use our system, an animator first provides phonetically annotated audio. The annotation may be done automatically [22], semi-automatically using a text transcript [22], or manually [34].

Trajectory synthesis is performed by Equation 17 using the trained phonetic models. This is done separately for the flow and the texture parameters. After the parameters are synthesized, Equation 7 from Section 6.3 is used to create the visual stream with the desired mouth movement. MMM synthesis takes on the order of about 7 seconds per frame for an image resolution of 624x472. The background compositing process adds on a few extra seconds of processing time. All times are computed on a 450 MHz Pentium II.

10. Evaluation

We have synthesized numerous examples using our system, spanning the entire range of 1-syllable words, 2-syllable words, short sentences, and long sentences. In addition, we have synthesized songs and foreign speech examples. Oue results may be viewed on the web at

`http://cerboli.mit.edu:8000/research/mary101/mary101.html`.

We evaluated our results by performing three different visual "Turing tests" to see whether human subjects can distinguish between real sequences and synthetic ones. Performance in all three experiments was close tochance level (50%) and not significantly different from it. Finally, we also evaluated our system by performing intelligibility tests in which subjects were asked to lip read a set of natural and synthetic utterances. Details on all experiments are described in [17].

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical fbw techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, pages 237–252, Santa Margherita Ligure, Italy, 1992.

[3] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, 1996.

[4] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. Technical Report 1431, MIT AI Lab, 1993.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[6] A. Black and P. Taylor. *The Festival Speech Synthesis System*. University of Edinburgh, 1997.

[7] M. Black, D. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding, Special Issue on Robust Statistical Techniques in Image Understanding*, pages 8–31, 2000.

[8] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In A. Rockwood, editor, *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 187–194, Los Angeles, 1999. ACM, ACM Press / ACM SIGGRAPH.

[9] M. Brand. Voice puppetry. In A. Rockwood, editor, *Proceedings of SIGGRAPH 1999*, Computer Graphics Proceedings, Annual Conference Series, pages 21–28, Los Angeles, 1999. ACM, ACM Press / ACM SIGGRAPH.

[10] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of SIGGRAPH 1997*, Computer Graphics Proceedings, Annual Conference Series, pages 353–360, Los Angeles, CA, August 1997. ACM, ACM Press / ACM SIGGRAPH.

[11] N. Brooke and S. Scott. Computer graphics animations of talking faces based on stochastic models. In *Intl. Symposium on Speech, Image Processing, and Neural Networks*, Hong Kong, April 1994.

[12] M. M. Cohen and D. W. Massaro. Modeling coarticulation in synthetic visual speech. In N. M. Thalmann and D. Thalmann, editors, *Models and Techniques in Computer Animation*, pages 139–156. Springer-Verlag, Tokyo, 1993.

[13] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, Freiburg, Germany, 1998.

[14] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1989.

[15] E. Cosatto and H. Graf. Sample-based synthesis of photorealistic talking heads. In *Proceedings of Computer Animation '98*, pages 103–110, Philadelphia, Pennsylvania, 1998.

[16] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic facial animation. In *Proceedings of SIGGRAPH 2002*, volume 21, pages 388–398, San Antonio, Texas, 2002.

[17] T. Ezzat, G. Geiger, and T. Poggio. Mary101:a trainable videorealistic speech animation. In G. B. . P. P. E. E. Vatiokis-Bateson, editor, *Audiovisual Speech Processing*. MIT Press, to appear.

[18] T. Ezzat and T. Poggio. Visual speech synthesis by morphing visemes. *International Journal of Computer Vision*, 38:45–57, 2000.

[19] F. Girosi, M. Jones, and T. Poggio. Priors, stabilizers, and basis functions: From regularization to radial, tensor, and additive splines. Technical Report 1430, MIT AI Lab, June 1993.

[20] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *Proceedings of SIGGRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 55–66, Orlando, FL, 1998. ACM, ACM Press / ACM SIGGRAPH.

[21] B. K. P. Horn and B. G. Schunck. Determining optical fbw. *Artifi cial Intelligence*, 17:185–203, 1981.

[22] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The SPHINX-II speech recognition system: an overview (http://sourceforge.net/projects/cmusphinx/). *Computer Speech and Language*, 7(2):137–148, 1993.

[23] M. Jones and T. Poggio. Multidimensional morphable models: A framework for representing and maching object classes. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, 1998.

[24] S. Y. Lee, G. Wolberg, and S. Y. Shin. Polymorph: An algorithm for morphing among multiple images. *IEEE Computer Graphics Applications*, 18:58–71, 1998.

[25] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proceedings of SIGGRAPH 1995*, Computer Graphics Proceedings, Annual Conference Series, pages 55–62, Los Angeles, California, August 1995. ACM, ACM Press / ACM SIGGRAPH.

[26] B. LeGoff and C. Benoit. A text-to-audiovisual-speech synthesizer for french. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Philadelphia, USA, October 1996.

[27] T. Masuko, T. Kobayashi, M. Tamura, J. Masubuchi, and K. Tokuda. Text-to-visual speech synthesis based on parameter generation from hmm. In *ICASSP*, 1998.

[28] F. I. Parke. *A parametric model of human faces*. PhD thesis, University of Utah, 1974.

[29] A. Pearce, B. Wyvill, G. Wyvill, and D. Hill. Speech and expression: A computer solution to face animation. In *Graphics Interface*, 1986.

[30] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of SIGGRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 75–84, Orlando, FL, 1998. ACM, ACM Press / ACM SIGGRAPH.

[31] T. Poggio and T. Vetter. Recognition and structure from one 2D model view: observations on prototypes, object classes and symmetries. Technical Report 1347, Artifi cial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

[32] S. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[33] K. Scott, D. Kagels, S. Watson, H. Rom, J. Wright, M. Lee, and K. Hussey. Synthesis of speaker facial movement to match selected speech sequences. In *Proceedings of the Fifth Australian Conference on Speech Science and Technology*, volume 2, pages 620–625, December 1994.

[34] K. Sjlander and J. Beskow. Wavesurfer - an open source speech tool. In *Proc of ICSLP*, volume 4, pages 464–467, Beijing, 2000.

[35] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, Dec 2000.

[36] G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

[37] K. Waters. A muscle model for animating three-dimensional facial expressions. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, volume 21(4), pages 17–24. ACM, July 1987.