MERL – A MITSUBISHI ELECTRIC RESEARCH LABORATORY
http://www.merl.com

Voice-driven animation

Matthew Brand and Ken Shan

TR-98-20 September 1998

Abstract

We introduce a method for learning a mapping between signals, and use this to drive facial animation directly from vocal cues. Instead of depending on heuristic intermediate representations such as phonemes or visemes, the system learns its own representation, which includes dynamical and contextual information. In principle, this allows the system to make optimal use of context to handle ambiguity and relatively long-lasting facial co-articulation effects. The output is a series of facial control parameters, suitable for driving many different kinds of animation ranging from photo-realistic image warps to 3D cartoon characters.

Publication History:–

1. 6jul98 first circulated.
2. 26aug98 accepted to Workshop on Perceptual User Interfaces, November 1998, San Francisco
3. 21sep98 final version uploaded to workshop web site.

Voice-driven animation

Matthew Brand and Ken Shan
MERL—a Mitsubishi Electric Research Lab
201 Broadway, Cambridge, MA, 02139
brand@merl.com

Abstract

We introduce a method for learning a mapping between signals, and use this to drive facial animation directly from vocal cues. Instead of depending on heuristic intermediate representations such as phonemes or visemes, the system learns its own representation, which includes dynamical and contextual information. In principle, this allows the system to make optimal use of context to handle ambiguity and relatively long-lasting facial co-articulation effects. The output is a series of facial control parameters, suitable for driving many different kinds of animation ranging from photo-realistic image warps to 3D *cartoon characters.*

1. From lip-syncing to facial animation

Psychologists and storytellers alike have observed that there is a good deal of mutual information between vocal and facial gesture [23]. Facial information can add significantly to the observer's comprehension of the formal [2] and emotional content of speech, and is considered by some a necessary ingredient of successful speech-based interfaces. Conversely, the difficulty of synthesizing believable faces is a widely-noted obstacle to producing acceptable digital avatars, agents, and animation. People are highly specialized for interpreting facial action; a poorly animated face can be disturbing and even can interfere with the comprehension of speech [18].

Lip-syncing, a large part of facial animation, is a laborious process in which the voice track is dissected (usually by hand) to identify features such as stops and vowels, then matching mouth poses are scheduled in the animation track, 2-10 per second. The overwhelming majority of lip-syncing systems are based on an intermediate phonemic representation, whether obtained by hand [19, 20], from text [7, 9, 1, 14] or, in the case of voice-keyed systems, via speech recognition [16, 24, 6]. Typically, phonemic or visemic tokens are mapped directly to lip poses, ignoring dynamical factors. Efforts toward dynamical realism have been limited. E.g., Video Rewrite [6] is a table-driven frame re-ordering system in which vocal (but not facial) co-articulation is partially modeled via triphones; Baldy [7] is a synthesized computer graphics head that uses an explicit vocal co-articulatory model derived heuristically from the psychological literature. Co-articulation is the interaction between nearby speech segments.

Phonemic and visemic representations are probably a suboptimal representation of the information common to voice and face, since they obliterate the relationships between vocal prosody and upper facial gesture, and between vocal and gesture energy. Moreover, there is inherent information loss in the discretization to phonemes. Attempts to generate lip poses directly from the audio signal (e.g., [17]) have been limited to predicting vowel shapes and ignore temporal effects.

None of these methods address the actual dynamics of the face. Facial muscles and tissues contract and relax at different rates. For example, there is co-articulation at multiple time-scales—50-300ms in the vocal apparatus, and longer on the face [15]. Furthermore, there is evidence that lips alone convey less than half of the visual information that human subjects can use to disambiguate noisy speech [2]. Much of the expressive and emotional content of facial gesture happens in the upper half of the face; this is not at all addressed by speech-oriented facial animation.

We propose a more direct mapping from voice to face by learning a model of the face's natural dynamics during speech, then learning a mapping from vocal patterns to facial motion trajectories. This strategy has several appealing properties: 1) Voice is analyzed with regard to learned categories of facial gesture, rather than with regard to hypothesized categories of speech perception. 2) Long-term dependencies such as facial co-articulations are implicitly modeled. 3) A probabilistic framework allows us to find the most probable face trajectory for a whole utterance, not just for a small window of time. 4) The output of the system is a set of facial control programs that can be used to drive 2D, 3D, or image-based face animations.

2. System overview

Figure 1 shows a block diagram of the system. We begin with a database of synchronized speech and video. We model facial dynamics (positions and velocities of facial features) with hidden Markov model (HMM), then split the HMM into two parts: a finite state machine which models the face's qualitative dynamics (e.g., expression to expression transition probabilities), and a set of Gaussian distributions

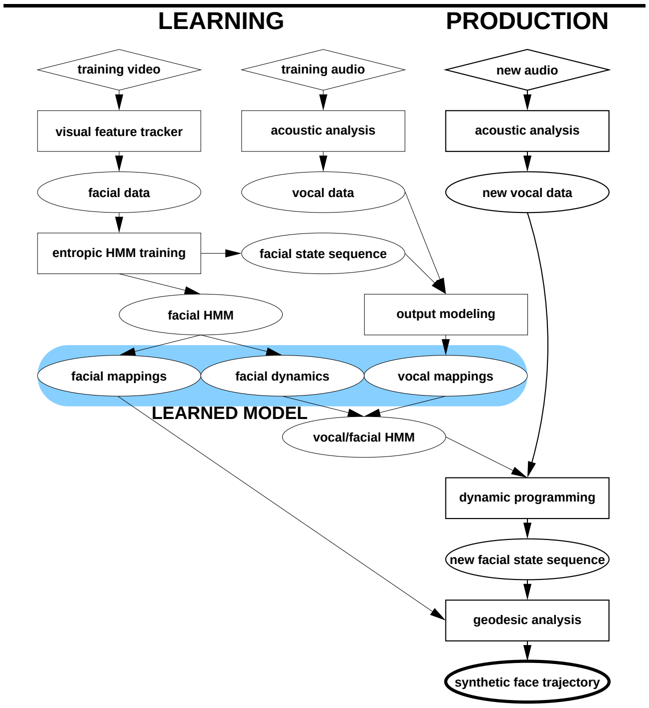


Figure 1. Block diagram of the learning and animation system. In the production pathway new audio is used to drive facial animation.

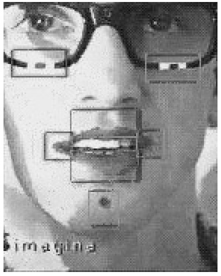


Figure 2. Tracking windows around five features on the lower face plus two registration points.

that associate those states to regions of facial configuration space. We then learn a second set of distributions—from regions of vocal configuration space to the states occupied by the face at the same time. This combines with the facial dynamical model to become a new HMM that analyzes new voice-tracks. Instead of giving us a most probable sequence of phonemes, it gives us a most probable sequence of facial states, using context from the full utterance for disambiguation when necessary. Using this sequence and the original set of facial output distributions, we solve for a maximally probable trajectory through facial configuration space. This is then used to drive the animation.

Two innovations make this scheme workable: 1) Given a state sequence, we have a closed solution for the maximally probable trajectory that mimics both the natural poses and velocities of the face (§2.3). 2) We can estimate probabilistic models which give us unique, unambiguous state sequences (§2.2). The second point is somewhat subtle: It is always possible to extract a most probable state sequence from an HMM via Viterbi analysis, but typically there may be thousands of other sequences that are only slightly less probable, so that the most probable sequence has only a tiny fraction of the total probability mass, and is a mediocre representation of the signal. We have developed a method for learning sparse HMMs by explicitly minimizing all forms of ambiguity (entropy); one consequence is that the Viterbi sequence typically has most of the probability mass.

2.1. Signal processing

To obtain facial articulation data, we developed a computer vision system that simultaneously tracks several individual features on the face, such as the corners of the mouth. Taking Hager's SSD texture-based tracker [11] as a starting point, we developed a mesh of such trackers to cover the face. Figure 2 shows an early example with 7 features; we are now tracking 25 points on the face. We assigned spring tensions to each edge connecting a pair of trackers, and the entire system was made to relax by simultaneously minimizing the spring energies and the residuals of the individual trackers. If a tracker "falls off" its landmark feature, spring forces from its neighbors tend to push it back into place. To estimate spring lengths and stiffnesses for a specific sequence, we run the video through the system, record the mean and variance of the distance between pairs of trackers, and use this to revise the spring properties. A few repetitions sufficed to obtain stable and accurate tracking in our training videos. Since obtaining accurate data was more important than stress-testing our tracker, we marked low-texture facial areas and asked subjects to reduce head motions.

To obtain a useful vocal representation, we calculate a mix of LPC and RASTA-PLP features [12]. These are known to be useful to speech recognition and somewhat robust to variations between speakers and recording conditions. Since these codings are designed for phonemic analysis, they aren't necessarily optimal indicators of facial activity. We are also experimenting with prosodic features such as the formants and the energy in sonorant frequency bands.

2.2. Learning

Our ultimate goal is to learn a mapping from the vocal features in a given frame to simultaneous facial features. The mapping is many-to-many: Many sounds are compatible with one facial pose; many facial poses are compatible with one sound. Were it not for this ambiguity, we could use a simple regression method such as a perceptron, neural network, or radial basis function network. Since much of the complexity arises from causal factors such as co-articulation, the best remedy is to use context from before and after the frame of interest. The fact that the disambiguating context has no fixed length or proximity to the current frame strongly recommends that we use a hidden Markov model, which (if properly trained) can make optimal use of context across an entire utterance, regardless of its length. An HMM uses its hidden states to carry contextual information forward and backward in time; training will naturally assign some states to that task.

Since the hidden state changes in each frame under the influence of the observed data, it is important for the matrix governing state transitions to be sparse, otherwise a context-carrying state will easily transition to a data-driven state, and the contextual information will be lost. We have developed a framework for training probabilistic models that minimizes their internal entropy; in HMMs that translates to maximizing compactness, sparsity, capacity to carry contextual information, and specificity of the states. The last property is particularly important because conventionally trained HMMs typically express the content of a frame as a mixture of states, making it impossible to say that the system was in any one state.

We briefly review the entropic training framework here, and refer readers to [3, 4] for details. We begin with a dataset $\boldsymbol{X}$ and a model whose parameters and structure are specified by the matrix $\boldsymbol{\theta}$. In conventional training, one guesses the sparsity structure of $\boldsymbol{\theta}$ in advance and merely re-estimates nonzero parameters to maximize the likelihood $P(\boldsymbol{X}|\boldsymbol{\theta})$. In entropic training, we learn the size of the $\boldsymbol{\theta}$, its sparsity structure, and its parameter values simultaneously by maximizing the posterior given by Bayes' rule,

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\boldsymbol{X}) \propto P(\boldsymbol{X}|\boldsymbol{\theta})P_e(\boldsymbol{\theta}) \tag{1}$$

where we define the entropic prior

$$P_e(\boldsymbol{\theta}) \propto e^{-H(\boldsymbol{\theta})} \tag{2}$$

and $H(\cdot)$ is an entropy measure defined on the model's parameters. Entropy measures uncertainty, thus we are seeking the least ambiguous model that can explain the data. The entropic prior can also be understood as a mathematization of Occam's razor: Smaller models are less ambiguous because they contain fewer alternatives. Interestingly, the prior itself can be derived from a logical proposition, and a transformation of the entropic posterior allows us to manipulate the Helmholtz free energy, giving deterministic annealing (a quasi-global optimization technique) and other connections to statistical physics [5].

Given a factorizable model such as an HMM, the maximum *a posteriori* (MAP) problem decomposes into a separate equation for each independent parameter θ_{ij}, each having its own entropic prior. We have found exact solutions for a wide variety of such equations, yielding very fast learning algorithms. MAP estimation extinguishes excess parameters and maximizes the information content of the surviving parameters. This allows us to learn the proper size and sparsity structure of a model. Frequently, entropic estimation of HMMs recovers a finite-state machine that is very close to the mechanism that generated the data.

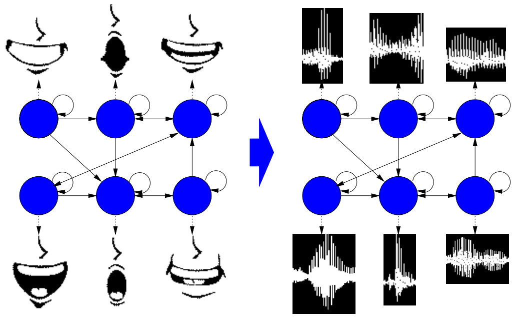


Figure 3. Reuse of the facial HMM's internal state machine in constructing the vocal HMM.

Using entropic estimation, we learn a facial dynamical model from the poses and velocities output by the vision system. We then use a dynamic programming analysis to find the most probable sequence of hidden states given the training video. Using this state sequence, we estimate output probabilities, given each state, of the synchronized audio track. This associates audio features to each facial state, resulting in a new vocal HMM which has the dynamics of the face, but is driven by the voice (Figure 3).

2.3. Synthesis

Given a new vocal track, we use the vocal HMM to find a maximally probable sequence of predicted facial states. Remember that this will follow the natural dynamics of the face, but it is steered by information in the new vocal track.

We then use the facial output probabilities to make a mapping from predicted states to actual facial configurations. Were we to simply pick the most probable configuration for each state—the mean face—the animation would jerk from pose to pose; the timing would be natural, but the jerkiness would not. Most phoneme- and viseme-based lip-sync systems solve this problem by interpolating or splining between poses. This might solve the jerkiness, but it is an *ad hoc* solution that degrades or destroys the natural timing.

A proper solution should yield a short, smooth trajectory that passes through regions of high probability density in configuration space at the right time. Prior approaches to trajectory estimation typically involve maximizing an equation having a probability term and penalty terms for excess length and/or kinkiness and/or point clumpiness. The user must choose a parameterization and weighting for each term. This leads to variational techniques that are often approximate, iterative, and computationally expensive (e.g., [22]). Moreover, the equation may have many local maxima and one may not be able to tell whether the found maxima is near-optimal or mediocre.

Our framework simplifies the problem so significantly that a closed-form solution is available. Because we model both pose and velocity, the facial output probabilities alone contain enough information to completely specify the smooth trajectory that is most consistent with the facial dynamics and a given facial state sequence.

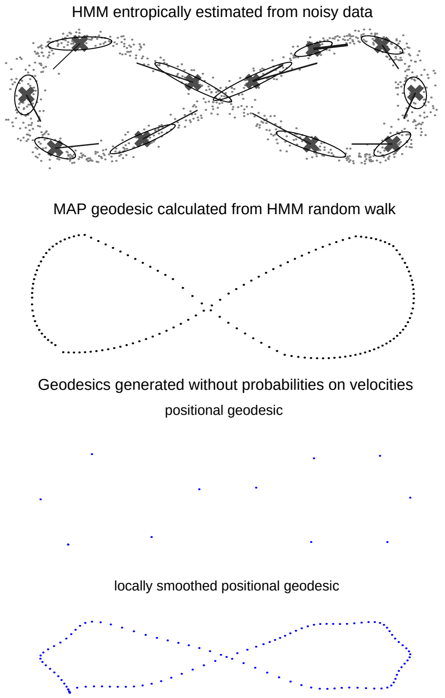
The formulation is quite clean: We assume that each state has Gaussian outputs that model positions and velocities. For simplicity of exposition, we'll assume a single Gaussian per state, but our treatment trivially generalizes to Gaussian mixtures. Let $\boldsymbol{\mu}_i, \dot{\boldsymbol{\mu}}_i$ be the mean position and velocity for state i,, and $\boldsymbol{K}_i$ be a full-rank covariance matrix relating positions and velocities in all dimensions. Furthermore, let $s(t)$ be the state governing frame t and let $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots\}$ be the variable of interest, namely, the points the trajectory passes through at frame 1,2,3,... Then we want the maximum of

$$\begin{aligned} \boldsymbol{Y}^* &= \operatorname*{argmax}_{\boldsymbol{Y}} \log \prod_t \mathcal{N}(\tilde{\boldsymbol{y}}_t; [\boldsymbol{\mu}_{s(t)}, \dot{\boldsymbol{\mu}}_{s(t)}], \boldsymbol{K}_{s(t)}) \\ &= \operatorname*{argmin}_{\boldsymbol{Y}} \sum_t \tilde{\boldsymbol{y}}_t \boldsymbol{K}_{s(t)}^{-1} \tilde{\boldsymbol{y}}_t^\top / 2 + c \qquad (3) \end{aligned}$$

where $\tilde{\boldsymbol{y}}_t = [\boldsymbol{y}_t - \boldsymbol{\mu}_{s(t)}, (\boldsymbol{y}_t - \boldsymbol{y}_{t-1}) - \dot{\boldsymbol{\mu}}_{s(t)}]$ is a row vector of instantaneous facial position and velocity. Eqn. 3 is a quadratic form having a single global maximum. Setting its derivative to zero yields a block-banded system of linear equations in which $\boldsymbol{y}_t$ depends only on $\boldsymbol{y}_{t-1}, \boldsymbol{y}_{t+1}$. For T frames and D dimensions, the system can be LU-decomposed and solved in time $O(TD^2)$ [10, §4.3.1]. To illustrate, Figure 4 shows an HMM entropically estimated from a noisy system that orbits in a figure-eight, and various ways of estimating trajectories from it.

2.4. Animation

The system synthesizes would-be facial tracking data—what probably would have seen had the training subject produced the input vocalization. This sequence of facial motion vectors can be used to control a 3D animated head model or to warp a 2D face image to give the appearance of motion. Or, by learning a mapping from tracking data back training video, we can directly synthesize new video. We chose a versatile solution which provides a surprisingly good illusion—a 2D image such as a photograph is texture-mapped onto a 3D model having a low triangle count. Simple deformations of the 3D model give a naturalistic illusion of facial motion while the shading of the image gives the illusion of smooth surfaces. The deformations can be applied directly by mapping control points to vertices in the model, or indirectly by mapping synthetic facial configurations to mixtures of Facial Action Units [8] that are defined on the model [21]. The latter approach has the advantage of giving us full 3D control of a model even when the training data is only 2D. Action units are also commonly used for facial animation and image coding, e.g., MPEG-4.



HMM entropically estimated from noisy data

MAP geodesic calculated from HMM random walk

Geodesics generated without probabilities on velocities

positional geodesic

locally smoothed positional geodesic

Figure 4. TOP: An entropically estimated HMM projected onto synthetic training data. An × indicates the mean output of a state; an ellipse indicates its covariance; and arcs indicate allowable transitions. SECOND: A trajectory generated using our method based on positional and velocity distributions. The state sequence is obtained from a random walk through the HMM. (Irregularities are due to variations between state dwells in the random walk.) THIRD: Traditionally, one solves for a geodesic using positional distributions, but this leads to all control points clumping on the means. BOTTOM: Clumpiness ameliorated by smoothing terms, but the trajectory is still unacceptable. (This could be improved if one is willing to hand-tune the objective function.)

3. Examples

We begin with a simple example. We obtained a 300-frame (12 seconds at 25 f.p.s.) quicktime movie of a French speaker saying roughly three sentences [13]. Since this sample is too small to capture much of the natural variation in vocal and facial gesture, we limited our efforts to extracting basic lip and jaw motions by tracking four points around the mouth and one on the jaw in 2D (see figure 2). Together with velocities, this produced a 20D observation vector. The voice observation vector consisted of 9 PLP bands, and velocities thereof. The face was modeled with an 8-state HMM. Because of the paucity of data and the high dimensionality

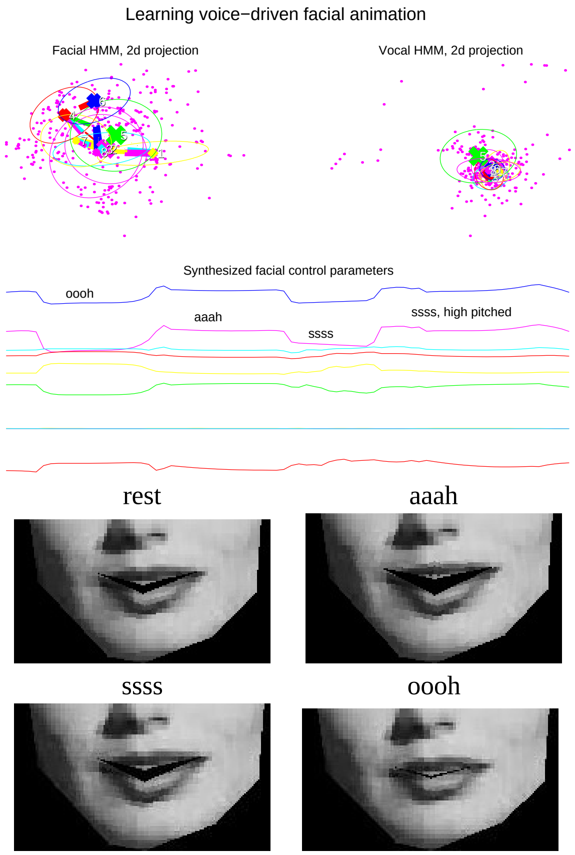


Figure 5. Synthesis of facial control parameters from a new voice-track. UPPER LEFT: The facial HMM superimposed on the facial data, projected down to 2D. UPPER RIGHT: The vocal HMM and data, similarly projected. MIDDLE: Time line of synthesized control parameters. BOTTOM: Selected frames from the resulting animation.

of the observations, we reduced the dimensionality of the data using PCA, preserving 98% of the variance. (300 points leaves 20 dimensions nearly empty, which can cause the Gaussians to collapse during learning.) Figure 5 shows results generated from novel audio of a few basic sounds: 'oooh', 'aaah', and a sibilant hiss 'ssss' made with different amounts of mouth opening.

The predicted face state sequence had an entropy rate of 0.0204. This means, roughly, that one out of every 34 states in the predicted sequence had a single plausible alternative. By contrast, a conventionally trained HMM yielded an entropy rate of 1.0296, roughly 1.8 plausible alternatives for *every* state in the predicted state sequence. Of that, the most probable sequence produced a substantially degraded animation, while the properly weighted combination of all such sequences produced only a slight improvement.

Even with quite modest training data, the animation generated from the entropically trained model produces the correct qualitative behavior for all the vowels and most consonantal stops. We turn now to larger training sets and quantitative measures of performance.

3.1. High quality coding with more learning

In a second set of experiments, we recorded subjects telling a variety of children's stories and processed a full 180 seconds of video, tracking 25 features on the face. Roughly 60 seconds of the data were modeled with a 24-state entropically estimated HMM. The perplexity (average branching factor) of the learned facial state machine was 2.19, indicating that the model is carrying context effects such as co-articulation an average of 4 frames (≈ 135 msec.) in either temporal direction, and that context can in principle be carried well over a second. Again, this is due to entropic estimation; an HMM conventionally trained from the same initialization carried context an average of just 1 frame.

Remarkably, we found that the training data could be quite accurately reconstructed (via the model) from its most probable state sequence. After string compression, this works out to facial motion coding of less than 4 bits per frame. Reconstruction of facial motion just from the vocal track was almost as good. We quantified this with a squared error measure of divergence between ground-truth ($\boldsymbol{x}$) and reconstructed ($\boldsymbol{y}$) facial motion vectors, weighted to strongly penalize motions in the wrong direction:

$$\mathrm{Err}(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y})^\top/(\boldsymbol{x}+\boldsymbol{y})(\boldsymbol{x}+\boldsymbol{y})^\top \qquad (4)$$

We reconstructed facial motion from 1) most probable state sequences of the ground truth motion; 2) the vocal track; and 3) a minimum squared error coding of the training set via action unit activations. The mean errors were:

reconstruction error from	training data	test data
most probable state sequence	0.1458	0.1783
vocal track	0.1882	0.2193
action unit coding	0.4735	0.4692

Note that synthesis from voice is significantly better than the reconstruction from action unit codings, indicating that the learned representation of the HMM is superior to the heuristic representation of the action units. The same ranking obtains if one switches to an unweighted squared-error measure.

Surprisingly, motion in the upper face was even more accurately predicted than motion around the mouth. One possible explanation is that upper face control is a much less complicated phenomenon, even though it seems less directly linked to vocal behavior.

Using the mean facial poses of the HMM states as guides, we defined a richer set of action units for a new and more detailed 3D model. Figure 6 shows this model animating Mt. Rushmore under the control of novel voice data.

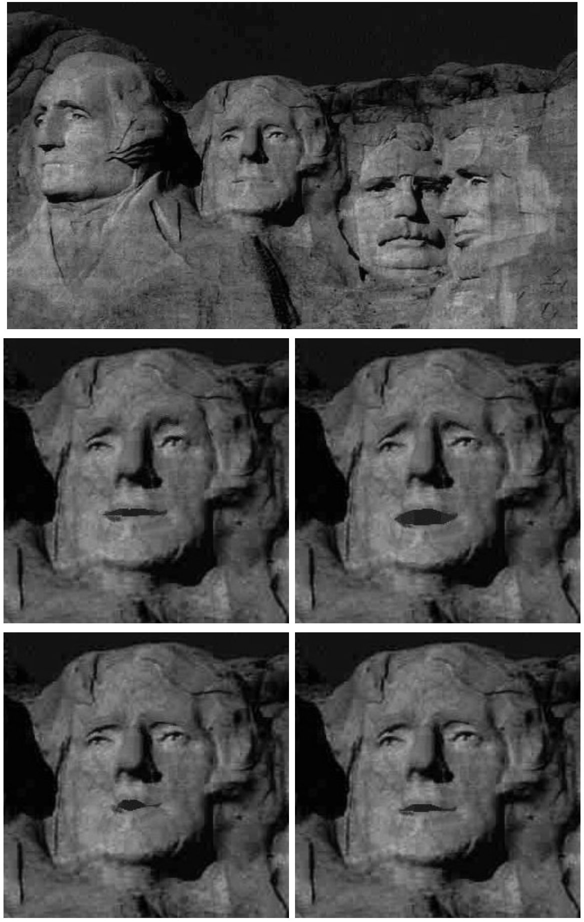


Figure 6. President Jefferson at rest (top) and face-syncing to novel audio. Note the eyebrows.

4. Conclusion

Voice-driven animation combines the voice, face, and facial mannerisms of three different people into a realistic speaking animation. The system stands on two innovations: An entropy-minimization algorithm learns extremely compact and accurate probabilistic models of facial behavior from training video; and a solution for geodesics extracts optimal facial motion sequences from hidden Markov models. Experiments show that, given novel audio, the system accurately generates lip and whole-face motions, even modeling subtle effects such as co-articulation.

5. Acknowledgments

The solution to the covariance-based linear equations was implemented by Ken Shan. The facial renderer was graciously provided to us by Robert Forchheimer, and modified to produce animations by Ilya Baran.

References

[1] J. E. Ball and D. T. Ling. Spoken language processing in the persona conversational assistant. In *ESCA Workshop on Spoken Dialogue Systems*, 1995.

[2] C. Benoit, C. Abry, M.-A. Cathiard, T. Guiard-Marigny, and T. Lallouache. Read my lips: Where? How? When? And so... What? In *8th Int. Congress on Event Perception and Action*, Marseille, France, July 1995. Springer-Verlag.

[3] M. Brand. Learning concise models of human activity from ambient video. Technical Report 97-25, Mitsubishi Electric Research Labs, November 1997.

[4] M. Brand. Structure discovery in conditional probability models via an entropic prior and parameter extinction. *Neural Computation (To appear.)*, October 1997.

[5] M. Brand. Pattern discovery via entropy minimization. In reviews for *1999 Workshop on Statistics and Artificial Intelligence*, June 1998.

[6] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Proc. ACM SIGGRAPH 97*, 1997.

[7] M. Cohen and D. Massaro. Modeling coarticulation in synthetic visual speech. In N. M. Thalmann and D. Thalmann, editors, *Models and Techniques in Computer Animation*. Springer-Verlag, 1993.

[8] P. Ekman and W. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Inc., Palo Alto, CA, 1978.

[9] T. Ezzat and T. Poggio. Miketalk: A talking facial display based on morphing visemes. In *Proceedings of the Computer Animation Conference*, June 1998.

[10] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins, 1996. 3rd edition.

[11] G. Hager and K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 1997. To appear. Available at www.cs.yale.edu/users/hager/papers.html.

[12] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, October 1994.

[13] F. Institute of Speech Communication, Grenoble. Leslevres quicktime video. http://ophale.icp.grenet.fr/Films.

[14] I. Katunobu and O. Hasegawa. An active multimodal interaction system. In *ESCA Workshop on Spoken Dialogue Systems*, 1995.

[15] Ken Stevens, MIT. Personal communication.

[16] J. Lewis. Automated lip-sync: Background and techniques. *The Journal of Visualization and Computer Animation*, 2:118–122, 1991. Available at www.idiom.com/˜zilla.

[17] D. F. McAllister, R. D. Rodman, and D. L. Bitzer. Speaker independence in lip synchronization. In *compugraphics '97*, December 1997.

[18] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[19] F. Parke. A parametric model for human faces. Technical Report UTEC-CSc-75-047, University of Utah, 1974.

[20] F. Parke. A model for human faces that allows speech synchronized animation. *Journal of Computers and Graphics*, 1(1):1–4, 1975.

[21] M. Rydfalk. CANDIDE, a parameterised face. Technical Report LiTH-ISY-I-0866, Department of Electrical Engineering, Linköping University, Sweden, October 1987. Demo available at http://www.bk.isy.liu.se/candide/candemo.html.

[22] L. Saul and M. Jordan. A variational principle for model-based interpolation. Technical report, MIT Center for Biological and Computational Learning, 1996.

[23] E. Walther. *Lipreading*. Nelson-Hall Inc, Chicago, 1982.

[24] K. Waters and T. Levergood. Decface: A system for synthetic face applications. *Multimedia Tools and Applications*, 1:349–366, 1995.