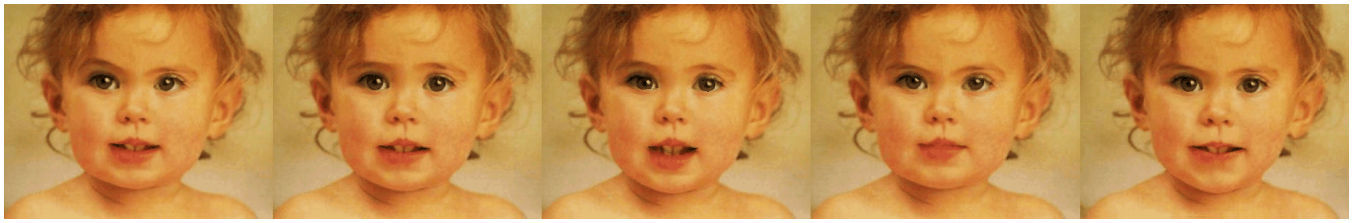
Voice Puppetry

Matthew Brand

MERL — a Mitsubishi Electric Research Laboratory
201 Broadway, Cambridge, MA 02139



Frames from a voice-driven animation, computed from a single baby picture and an adult model of facial control. Note the changes in upper facial expression. See figures 5, 6 and 7 for more examples of predicted mouth shapes.

Abstract

We introduce a method for predicting a control signal from another related signal, and apply it to *voice puppetry*: Generating full facial animation from expressive information in an audio track. The voice puppet learns a facial control model from computer vision of real facial behavior, automatically incorporating vocal and facial dynamics such as co-articulation. Animation is produced by using audio to drive the model, which induces a probability distribution over the manifold of possible facial motions. We present a linear-time closed-form solution for the most probable trajectory over this manifold. The output is a series of facial control parameters, suitable for driving many different kinds of animation ranging from video-realistic image warps to 3D cartoon characters.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.2.9 [Artificial Intelligence]: Robotics—Kinematics and Dynamics; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Time-varying images; G.3 [Mathematics of Computing]: Probability and Statistics—Time series analysis; E.4 [Data]: Coding and Information Theory—Data compaction and compression; J.5 [Computer Applications]: Arts and Humanities—Performing Arts

Keywords: Facial animation, lip-syncing, control, learning, computer vision and audition.

1 Face-syncing and control

As rendering techniques begin to deliver realistic-looking scenes, people are beginning to expect realistic-looking behavior. Therefore control is an increasingly prominent problem in animation. This is especially true in character facial animation, where good lip-syncing and dynamic facial expression are necessary to make a character look lively and believable. Viewers are highly attentive to facial action, quite sophisticated in their judgments of realism, and easily distracted by facial action that is inconsistent with the voice track.

We introduce methods for learning a control program for speech-based facial action from video, then driving this program with an audio signal to produce realistic whole-face action, including lip-syncing and upper-face expression, with correct dynamics, co-articulation phenomena, and ancillary deformations of facial tissues. In principle this method can be used to reconstruct any process from a related signal, for example, to synthesize dynamically correct 3D body motion from a sequence of shadows. We demonstrate with facial animation because of the obvious complexity of the process being modeled—the human face has many degrees of freedom, many nonlinear couplings, and a rather complicated control program, the mind.

Voice puppetry provides a low-cost quick-turnaround alternative to motion capture, with the additional flexibility that an actor's facial manner can be re-used over and over again to "face-sync" completely novel audio by other speakers and at other frame rates. It is fully automatic but an animator can intercede at any level to add detail. All algorithms have time complexity linear in the length of the input sequence; production time is slightly faster than utterance-time on a contemporary mid-level workstation.

2 Background

Psychologists and storytellers alike have observed that there is a good deal of mutual information between vocal and facial gesture [27]. Facial information can add significantly to the observer's comprehension of the formal [3] 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 and animations. People are highly specialized for interpreting facial action; a poorly animated face can be disturbing and even can interfere with the comprehension of speech [20].

Lip-syncing alone is a laborious process: The voice track is dissected (often by hand) to identify features such as stops and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGGRAPH 99, Los Angeles, CA USA
Copyright ACM 1999 0-201-48560-5/99/08 . . . $5.00

vowels, then matched mouth poses are scheduled in the animation track, 2-10 per second. Because it can overwhelm production schedules, lip-syncing has been the focus of many attempts at quasi-automation. Nearly all lip-syncing systems are based on an intermediate phonemic representation, whether obtained by hand [23, 24], from text [9, 12, 1, 17] or, with varying degrees of success, via speech recognition [18, 28, 7, 8, 29]. Typically, phonemic or visemic tokens are mapped directly to lip poses, ignoring dynamical factors. Efforts toward dynamical realism have been heuristic and use limited contextual information (e.g., [10, 7]). Consider the problem of co-articulation—the interaction between nearby speech segments due to latencies in tissue motion. To date, all attempts at co-articulation have depended on heuristic formulæ and *ad hoc* smoothing methods. E.g., BALDY [9] is a phoneme-driven computer graphics head that uses hand-designed vocal co-articulatory models inspired by the psychological literature. Although VIDEO REWRITE [7] works by re-ordering existing video frames rather than by generating animations, it deserves mention because it partially models vocal (but not facial) co-articulation with triphones—phonemes plus one 1 unit of left and right context. The quality of a video rewrite is determined by the amount of video that is available to be provide triphone examples and how successfully it is analyzed; smoothing is necessary because triphones don't fully constrain the solution and no video will provide an adequate stock of triphones.

Considerable information can be lost when discretizing to phonemic or visemic representations. The international phonetic alphabet is often mistaken for a catalog of the sounds and articulations humans make while communicating; in fact, phonemes are designed only to provide the acoustic features thought necessary to distinguish pronunciations of higher, meaning-carrying language elements. Phonemic representations are useful for analysis but quite suboptimal for synthesis because they obliterate predictive relationships such as those of vocal prosody to upper facial gesture, vocal energy to gesture magnitude, and vocal phrasing to lip articulation. There have been attempts to circumvent phonemes and generate lip poses directly from the audio signal (e.g., [22, 19]) but these are limited to predicting instantaneous vowel shapes.

None of these methods address the actual dynamics of the face. Facial muscles and tissues contract and relax at different rates. There are co-articulations at multiple time-scales—50-250 milliseconds in the vocal apparatus [21], possibly longer on the face. Furthermore, there is evidence that lips alone convey less than half of the visual information that human subjects can use to disambiguate noisy speech [3]. Much of the expressive and emotional content of facial gesture occurs in the upper half of the face. This is not addressed at all in speech-driven systems; some text-driven systems attempt upper-face animation, usually via hand annotations or *ad hoc* rules that exploit clues to sentence meaning such as punctuation.

We propose a more realistic mapping from voice to face by learning a model of a face's *observed* dynamics during speech, then learning a mapping from vocal patterns to facial motion trajectories. Animation is accomplished by using voice information to steer the model. This strategy has several appealing properties:

- Voice is analyzed with regard to learned (equivalently, optimized) categories of facial gesture, rather than with regard to hypothesized categories of speech perception.
- A consistent probabilistic framework allows us to find the optimal face trajectory for a whole utterance, making full use of forward and backward context and avoiding unjustified shortcuts such as smoothing or windowing.
- Video is analyzed just once, for training; the resulting model can be used re-used to face-sync any other person or creature to novel audio.
- The puppet animates speech and non-speech sounds.
- It predicts full facial motion from the neck to the hairline.
- The output is a sequence of facial motion vectors that can be used to drive 2D, 3D, or image-based animations.

3 Modeling the facial behavior manifold

It is useful to think of control in terms of the face's true behavioral manifold—a surface of all possible facial pose and velocity configurations embedded in a high-dimensional measurement space, like crumpled paper in 3-space. Actual performances are trajectories over this manifold. Our learning strategy is to piecewise approximate this manifold with quasi-linear submanifolds, then glue together these pieces with transition probabilities. Approximation is unavoidable because there isn't enough information in a finite training set to determine the shape of the true manifold. Therefore our control model is a probabilistic finite state machine, in which each state has an "output" probability distribution over facial poses and velocities, including how they covary. E.g., for every instantaneous pose each state predicts a unique most likely instantaneous velocity. The states are glued together with a distribution of transition probabilities that specify state-to-state switching dynamics and, implicitly, expected state durations.

Formally, this is a *hidden Markov model* (HMM)—Markov because all the context needed to do inference can be summed up in a vector of current state probabilities, and hidden because we never actually observe the states; we must infer them from the signal. For this we have the Viterbi algorithm [13], which identifies the most likely state sequence given a signal. The related Baum-Welch algorithm [2] estimates parameter values, given training data and a prior specification of the model's finite state machine. Both algorithms are based on dynamic programming and give locally optimal results in linear time. For this reason, HMMs dominate the literature of speech and gesture recognition.

Unfortunately, even for very small problems such as individual phoneme recognition, finding an adequate state machine is a matter of guesswork. This limits the utility of HMMs for more complex modeling because the structure of the state machine (pattern of available transitions) is the most important determinant of a model's success. Structure also determines the machine's ability to carry context; the rate at which an HMM forgets context is determined by how easily it can transition between any two states.

Voice puppetry features two significant innovations in the theory of HMMs: In §4.2 we outline the mathematical basis for training algorithms that estimate both the HMM parameter values *and* the structure of its finite state machine. This substantially generalizes Baum-Welch. In §4.5 we introduce an efficient solution for synthesizing the most probable signal from a state sequence. This can be thought of as an inverse Viterbi. Together, these techniques allow us to turn HMMs—traditionally only good enough for classification—into models of behavioral manifolds that are accurate enough for synthesis. As such, they can be trained to predict any nonrandom time-varying signal from a coordinated signal.

4 System overview

Figure 1 schematically outlines the main phases of voice puppetry. In **training** (first line), video is analyzed to yield a probabilistic finite state machine; a mapping from states into regions of facial configuration space; and an occupancy matrix giving state probabilities for each frame of the training sequence. In

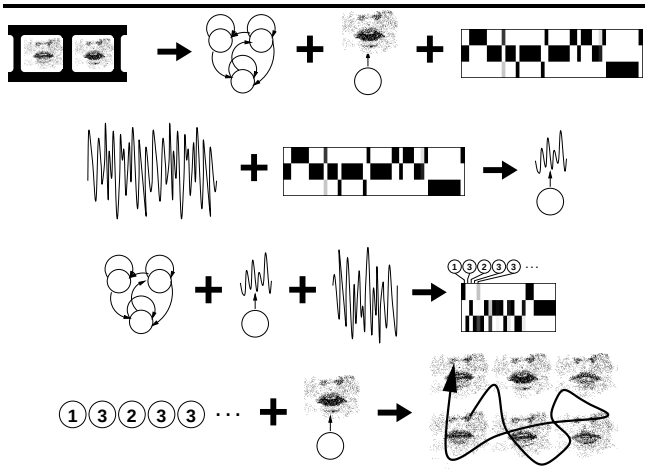
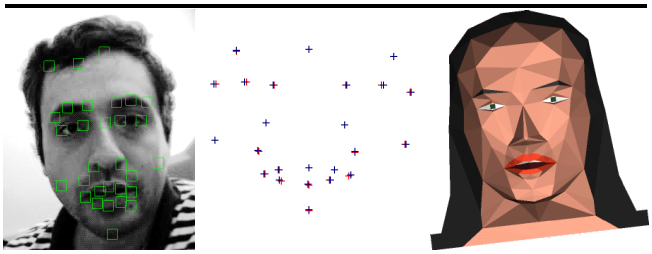


Figure 1: Schematic of the training, remapping, analysis, and synthesis steps of voice puppetry. See §4 overview.



A. tracking B. reconstruction C. control

Figure 2: (A) Tracking of facial features for training data. See §4.1. (B) Voice-predicted facial feature locations (blue) superimposed over ground-truth locations (red). See §4.5. (C) A 3D model articulated via vertex motions. See §4.6.

remapping (second line), the occupancy matrix is combined with the synchronized audio to give each state a dual mapping into acoustic feature space. These two steps define a puppet. In **analysis** of novel audio (third line), the state machine and the vocal distributions are combined to form an HMM which is used to analyze control audio, resulting in a most likely facial state sequence. This is much like speech recognition, except that the units of interest are facial states rather than phonemes. In **synthesis** (last line), the system solves for a trajectory through facial configuration space that is optimal with regard to the state sequence and learned facial output distributions.

4.1 Signal processing

To obtain facial articulation data, we developed a computer vision system that simultaneously tracks many individual features on the face, such as corners and creases of the lips. Taking Hager's SSD texture-based tracker [15] as a starting point, we developed a mesh of such trackers to cover the face. Figure 2A shows the system tracking 26 points on a 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 re-estimate the spring properties. A few repetitions sufficed to obtain stable and accurate tracking in our training videos. By tracking from two views we can also obtain stereo estimates of 3D depth. Our tracker can track on unmarked faces but depends on high-quality video to deliver facial texture, e.g., wrinkles or beard shadow. 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 audio features [16]. These are known to be useful to speech recognition and somewhat robust to variations between speakers and recording conditions. However, they are designed for phonemic analysis and aren't necessarily good indicators of facial activity. We also extract some prosodic features such as the formant frequencies and the energy in sonorant frequency bands.

Note that the puppet may work equally well with other representations of vocal and facial signals, and even different kinds of signals. Indeed, the success of our experiments notwithstanding, "Where in the signal is the information?" is still an open question for voice-driven facial animation and more generally for face perception and speech recognition.

4.2 Learning by entropy minimization

The mapping from vocal configurations to facial configurations 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 regression method such as a 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; a sufficiently powerful training algorithm 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 probability 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 also 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 [5, 4] for details and derivations. We begin with a dataset $\boldsymbol{X}$ and a model whose parameters and structure are specified by the vector $\boldsymbol{\theta}$. In conventional training, one guesses the sparsity structure of $\boldsymbol{\theta}$ in advance and merely re-estimates nonzero parameters to maximize the likelihood function $f(\boldsymbol{X}|\boldsymbol{\theta})$. In entropic training, we learn the size of the $\boldsymbol{\theta}$, its sparsity structure, and its parameter values simultaneously by maximizing the posterior probability given by Bayes' rule,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[P(\boldsymbol{\theta}|\boldsymbol{X}) \propto f(\boldsymbol{X}|\boldsymbol{\theta}) P_e(\boldsymbol{\theta}) \right] \tag{1}$$

Bayes' rule tells us how the probability of a hypothesis $\boldsymbol{\theta}$ changes after we have seen some evidence $\boldsymbol{X}$. The key to entropic estimation is that we derive the prior probability of a hypothesis

from its entropy,

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

where $H(\cdot)$ 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 be understood as a mathematization of Occam's razor: Choose the simplest hypothesis that adequately explains the data. Simpler models are less ambiguous because they allow fewer alternatives. Entropic estimation has interpretations as optimal compression and free energy minimization [5], depending on the formulation of $H(\boldsymbol{\theta})$. The free energy interpretation derives from the differential entropy $H_{\mathcal{F}}(\boldsymbol{\theta}) = -\int P(\boldsymbol{X}|\boldsymbol{\theta}) \log P(\boldsymbol{X}|\boldsymbol{\theta})\ d\boldsymbol{X}$ or the more tractable expected entropy; algorithmic complexity theory recommends that we use the compression formulation, which upper-bounds $H_{\mathcal{F}}$ with the sum the entropies of the model's component distributions. For discrete-state Gaussian-output HMMs, the likelihood function and compression entropy are:

$$f(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\mathcal{S}} \prod_{t}^{T} \theta_{s(t)|s(t-1)} \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{s(t)}, \boldsymbol{K}_{s(t)}), \tag{3}$$

$$H_{\mathcal{C}}(\boldsymbol{\theta}) = \sum_{i} \sum_{j} \theta_{j|i} \log 1/\theta_{j|i} + \tfrac{1}{2} \log(2\pi e)^d |\boldsymbol{K}_i|, \tag{4}$$

where $\theta_{j|i}$ are transition probabilities and $\boldsymbol{\mu}_i, \boldsymbol{K}_i$ are the d-dimensional mean and covariance of the i^{th} state's Gaussian output probability density function.

Given a factorizable model such as an HMM, the maximum *a posteriori* (MAP) problem decomposes into a separate equation for each independent parameter, each having its own entropic prior. In [5] we present exact solutions for a wide variety of such equations, yielding very fast learning algorithms; the case of HMMs is extensively treated in [4]. MAP estimation extinguishes excess parameters and maximizes the information content of the surviving parameters. Consequently, if we begin with a large fully-connected HMM, the training procedure whittles away all parts of the model that are not in accord with the hidden structure of the signal. 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.

4.2.1 Example

The topmost illustration in figure 4 shows an HMM entropically estimated from very noisy samples of a system that orbits in a figure-eight. The true system is a 2D manifold (phase and its rate of change) embedded in a 4D measurement space (observed 2D position and velocity); the HMM approximates this manifold with neighborhoods of locally consistent curvature in which velocity covaries linearly with position. Note that even though the data is noisy and has a continuation ambiguity where it crosses itself, the entropically estimated HMM recovers the deterministic structure of the system. A conventionally estimated HMM will get "lost" at the crossing, bunching states at the ambiguity and leaving many of them incorrectly over-connected, thus allowing multiple circuits on either loop as well as small circuits on the crossing itself. It is this additional concision and precision of entropic models makes them significantly outperform their conventionally estimated counterparts in traditional inference tasks, and enables novel applications such as voice puppetry.

4.3 Training and remapping

Using entropic estimation, we learn a facial dynamical model from the time-series of poses and velocities output by the vision system.

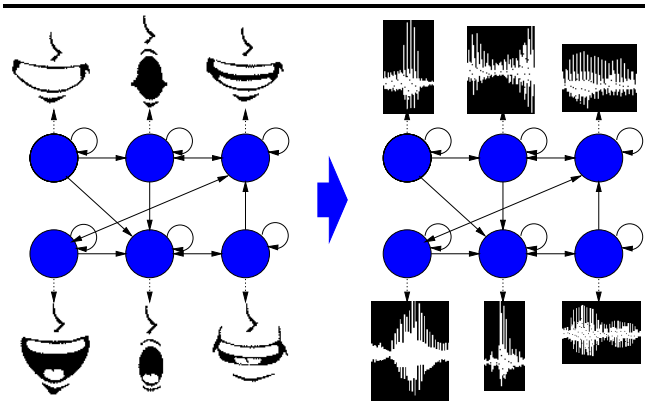


Figure 3: Reuse of the facial HMM's internal state machine in constructing the vocal HMM. Circles signify hidden states; arrows signify conditional probabilities; icons signify regions of facial and vocal configuration space contained within each output distribution. See §4.3.

The learning algorithm gives us an HMM plus an occupancy matrix $\gamma_{i,t}$ = Prob(HMM hidden state i explains frame t) of the training video. We use γ to estimate a second set of 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).

4.4 Analysis

Given a new vocal track, we apply the Viterbi algorithm to the vocal HMM to find the most likely sequence of predicted facial states. Although it is steered by information in the new vocal track, the Viterbi sequence is constrained to follow the natural dynamics of the face.

4.5 Synthesis

We use the facial output probabilities to make a mapping from the Viterbi states to actual facial configurations. Were we to simply pick the most probable configuration for each state—its mean face—the animation would jerk from pose to pose. Most phoneme- and viseme-based lip-sync systems address this problem by interpolating or splining between poses. This might ameliorate the jerkiness, but it is an *ad hoc* solution that ignores the face's natural dynamics.

A proper solution should yield a short, smooth trajectory that passes through regions of high probability density in configuration space at the right time—in our framework, a geodesic on the facial behavior manifold. Prior approaches to trajectory estimation typically involve optimizing an objective function having a likelihood term plus 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 algorithms that are often approximate and computationally intensive (e.g., [26]); often the objective function is nonconvex and one cannot tell whether the found optimum is global or mediocre.

Our current setting constrains the problem so significantly that a globally optimal 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 $\boldsymbol{K}_i^{-1} = \begin{bmatrix} \boldsymbol{K}_i^{xx} & \boldsymbol{K}_i^{x\dot{x}} \\ \boldsymbol{K}_i^{\dot{x}x} & \boldsymbol{K}_i^{\dot{x}\dot{x}} \end{bmatrix}$ 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 $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots\}^\top$ be the variable of interest, namely, the points the trajectory passes through at frame 1,2,3,... (All vectors in this paper are row-major.) We seek the maximum likelihood trajectory

$$\begin{aligned} \boldsymbol{Y}^* &= \underset{\boldsymbol{Y}}{\operatorname{argmax}} \log \prod_t \mathcal{N}(\tilde{\boldsymbol{y}}_t; \boldsymbol{K}_{s(t)}) \\ &= \underset{\boldsymbol{Y}}{\operatorname{argmin}} \sum_t \tilde{\boldsymbol{y}}_t \boldsymbol{K}_{s(t)}^{-1} \tilde{\boldsymbol{y}}_t^\top / 2 + c \end{aligned} \tag{5}$$

where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{K})$ is the Gaussian probability of $\boldsymbol{x}$ given covariance $\boldsymbol{K}$; and $\tilde{\boldsymbol{y}}_t = [\boldsymbol{y}_t - \boldsymbol{\mu}_{s(t)}, (\boldsymbol{y}_t - \boldsymbol{y}_{t-1}) - \dot{\boldsymbol{\mu}}_{s(t)}]$ is a vector of the mean-subtracted facial position and velocity at time t. Eqn. 5 is a quadratic form having a single global optimum. Setting its derivative to zero yields a block-banded system of linear equations:

$$\begin{bmatrix} \boldsymbol{K}_{s(t)}^{xx} + \boldsymbol{K}_{s(t)}^{\dot{x}\dot{x}} \\ \boldsymbol{K}_{s(t)}^{\dot{x}x} + \boldsymbol{K}_{s(t)}^{\dot{x}\dot{x}} \\ \boldsymbol{K}_{s(t+1)}^{\dot{x}x} + \boldsymbol{K}_{s(t+1)}^{\dot{x}\dot{x}} \\ \boldsymbol{K}_{s(t+1)}^{\dot{x}x} \\ \boldsymbol{K}_{s(t+1)}^{\dot{x}\dot{x}} \end{bmatrix}^{\perp} \begin{bmatrix} \boldsymbol{y}_t - \boldsymbol{\mu}_{s(t)} \\ \boldsymbol{y}_t - \boldsymbol{y}_{t-1} - \dot{\boldsymbol{\mu}}_{s(t)} \\ \boldsymbol{y}_t - \boldsymbol{y}_{t+1} \\ \boldsymbol{\mu}_{s(t+1)} \\ \dot{\boldsymbol{\mu}}_{s(t+1)} \end{bmatrix}^{\perp\top} = \boldsymbol{0} \tag{6}$$

where the block-transpose $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{\perp} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{C} \\ \boldsymbol{B} & \boldsymbol{D} \end{bmatrix} \neq \begin{bmatrix} \boldsymbol{A}^\top & \boldsymbol{C}^\top \\ \boldsymbol{B}^\top & \boldsymbol{D}^\top \end{bmatrix}$ and $\boldsymbol{K}_i^{\dot{x}x} = (\boldsymbol{K}_i^{x\dot{x}} + \boldsymbol{K}_i^{\dot{x}x})/2$. For T frames and $D = \dim(\boldsymbol{y}_t)$ dimensions, the system can be LU-decomposed and solved in time $O(TD^3)$ [14, §4.3.1]. By scaling the velocity terms, one may also solve for this trajectory at frame rates other than that of the training video.

Figure 4 shows various ways of estimating trajectories from an HMM model of the manifold of motion in a figure-eight. Increasing the number of HMM states improves the quality of the synthesized trajectory, provided there is sufficient data to support estimates of the additional parameters. Entropic estimation will automatically remove insufficiently supported parameters.

Eqn. 5 is only justified when the Viterbi sequence $\mathcal{V} = \{s(1), s(2), \ldots, s(T)\}$ strongly dominates the distribution of probable sequences. The Viterbi sequence, while most likely, may only represent a small fraction of the total probability mass—there may be thousands of slightly different state sequences that are nearly as likely. If this were to happen in the voice puppet, $\mathcal{V}$ would be a very poor representation of the relevant information in the audio, and the animation quality would suffer greatly. Consequently, we found that voice puppetry works very poorly with conventionally estimated HMMs. These problems are virtually banished with entropically estimated models because entropy minimization concentrates the probability mass on the optimal Viterbi sequence. (see §5, paragraph 2 for an empirical example). In [6] we present a full Bayesian MAP solution which considers *all* possible state sequences and show that it and the maximum likelihood solution are only valid for low-entropy models, where they give almost identical results inferring 3D full-body pose and motion from shadows.

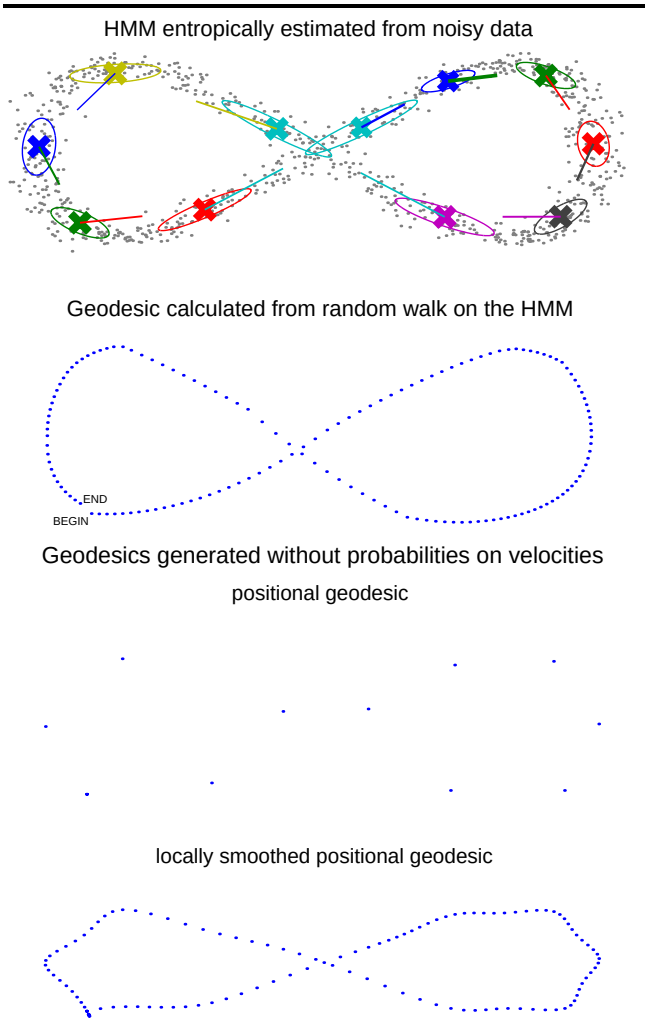


Figure 4: TOP: An entropically estimated HMM projected onto synthetic training data. An × indicates the mean output of a state; an ellipse indicates its covariance; and arcs indicate allowable transitions. See §4.2.1. SECOND: A trajectory generated using our method based on positional and velocity distributions. The state sequence is obtained from a random walk through the HMM. (Irregularities are due to variations between state dwells in the random walk.) See §4.5. THIRD: If one solves for a geodesic using just positional constraints, all control points clump on the means. BOTTOM: Traditionally, clumpiness is ameliorated by smoothing terms, but the trajectory is still unacceptable. (This could be improved if one is able and willing to hand-tune the objective function.)

4.6 Animation

The puppet synthesizes would-be facial tracking data—what most likely would have been seen had the training subject produced the input vocalization. This 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 an inverse mapping from tracking data back to 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—roughly 200 (figure 2C). Deformations of the 3D model give a naturalistic illusion of facial motion while the smooth shading of the image gives the illusion of smooth surfaces. The deformations can be applied directly by moving vertices according to puppet output, or indirectly by projecting synthesized facial configurations onto a basis set of

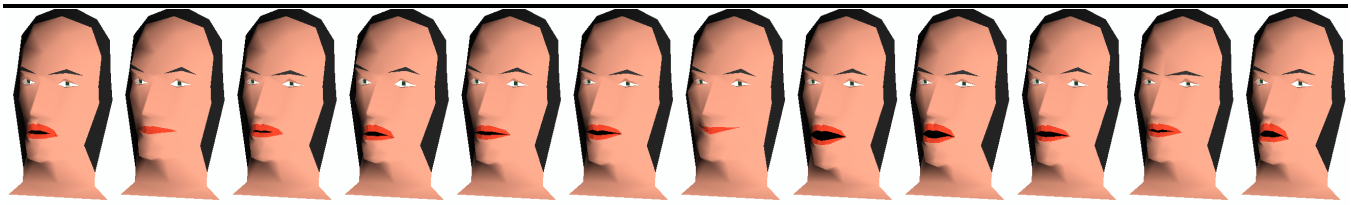


Figure 5: Visualization of the mean configuration for some of the learned states. Dynamical content is not shown.

motion vectors (a.k.a. facial action units) that are defined on the model (e.g., [25]). 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.

5 Examples

We recorded subjects telling a variety of children's stories and processed 180 seconds of video, tracking 25 features on the face, mostly around the mouth and eyes. Roughly 60 seconds of the data were modeled with a 26-state entropically estimated HMM. Many of the learned states had mean outputs resembling visemes and common facial morph targets, augmented with dynamical content (figure 5). The perplexity (average branching factor) of the learned facial state machine was 2.08, indicating that the model is carrying context effects such as co-articulation an average of ≈ 4.5 frames ($\approx$ 150 milliseconds) in either temporal direction[1]. In practice, we have seen this model carry context over 330 milliseconds, indicating that the system has discovered facial co-articulation phenomena that last longer than vocal co-articulations (and have yet to be mentioned in the speech psychology literature). These properties are due to entropic estimation; an HMM conventionally trained from the same initialization carried context an average of slightly under 2 frames.

Figure 6 shows this model animating Mt. Rushmore under the control of novel voice data. In this synthesis task, the predicted face state sequence had an entropy rate of 0.0315. This means that roughly one out of every 22 predictions in the facial state sequence had a single plausible alternative. By contrast, a conventionally trained HMM yielded an entropy rate of 0.875—roughly 2.4 plausible alternatives for *every* prediction in the facial state sequence. As expected, the most probable sequence from the conventionally estimated HMM yielded an unacceptably degraded animation, while the properly weighted combination of all such sequences produced only a slight improvement.

5.1 Evaluation via error and coding measures

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 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 penalize motions in the wrong direction:

$$\mathrm{Err}(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y})^\top / (\boldsymbol{x}+\boldsymbol{y})(\boldsymbol{x}+\boldsymbol{y})^\top \qquad (7)$$

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 data via activations of action units of the facial action coding system (FACS[2] [11, 25]). The table below shows mean errors as well as coding costs for storing and transmitting animations:

coding	model Kbytes	bits/ frame	reconstruction error train	test
state sequence	≈ 1.1	≈ 4	0.1255	0.1698
vocal features	≈ 2.0	< 500	0.1731	0.2115
FACS	≈ 0.6	> 600	0.4735	0.4692

The same ranking obtains if one switches to an unweighted squared-error measure. 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 psychologically-motivated but heuristic representation of FACS.

We obtained even better results by training and using separate models for the lower and upper face (e.g., eyes and up). Surprisingly, even with a single model, motion in the upper face is more accurately predicted than motion around the mouth. One possible explanation is that upper facial behavior is a much less complicated phenomenon, even though it seems less directly linked to vocal behavior.

5.2 Evaluation by naive viewers

In order to judge the subjective quality of the animations, we designed a set of blind trials in which naive observers tried to distinguish synthesized from real facial motion. We took 1500 frames of tracked video that had not been used for training, set the tracking data aside, and synthesized new facial motion from the audio. We generated animations from both the synthesized motion and the "ground-truth" tracked motion, broke each animation into three segments, and presented all segments in random order to naive observers. The subjects were asked to select the "more natural" animations. Three observers consistently preferred the synthesized animation; three consistently preferred the ground-truth animation; and one preferred the ground-truth animation in two out of three segments. This modest experiment indicates that while true and synthesized facial action can be distinguished (real facial action is more varied); they are almost equally plausible to naive viewers.

6 Discussion

The main determinant of puppetry quality is the extent and variety of speech behavior in the training video. With 12 seconds of

[1] More precisely, $\log_{2.08} 26 \approx 4.5$ is the average number of transitions the state machine takes to go between any two states. We use this as a heuristic indicator of the model's memory. The actual amount of time the HMM takes to forget that it was in any particular state is a function of the data and the output distributions, and can be determined empirically from differences between the most likely state sequence and the set of states that are most likely to output the observed data.

[2] FACS, like phonemes and visemes, was designed for psychological analysis, but has been pressed into service for modeling and coding by computer scientists.

Figure 6: President Jefferson at rest and face-syncing to novel audio. Animation runs from neck to hairline. On contemporary hardware, compute time is less than the duration of the utterance.

training video we can produce tolerable animation; with 3 minutes we approach video-realism. The quality of puppeteering degrades gracefully as we increase acoustic noise levels or change to microphones or speakers unlike those in the training set. E.g., when trained on adult men, the puppet has some difficulty with children's and women's voices. We would recommend separate models for each group because of large differences in facial manner and spectral profiles between gender and age groups. It is possible to train one large model on all groups, but this requires more data than needed for separate models.

We have used French-trained puppets to produce English animations and English-trained puppets to produce Russian and Japanese animations. This compares favorably with phoneme-based systems, which typically use an English subset of phonemes. We have also found it reasonably easy, via projection, to animate heads with substantially varied geometries, e.g., toddlers and animals (figure 7).

We currently train with NTSC 29.97Hz video—a sampling rate too low to reliably capture fast facial transients such as plosives and blinks. The puppet can infer most plosives from context, but true film-quality puppetry will probably require higher-resolution training data, tracking a hundred or so points on the face over tens of minutes of 100Hz video. In addition, we currently make no effort to track and model the shape of the tongue; we are currently looking into using archival x-ray films to complement the training set. Finally, for photo-realism we must handle wrinkling and changes in skin translucency; we are exploring variants of voice puppetry that predict changes in both the facial geometry and the texture map.

The voice puppet is fully automatic. Animators, on the other hand, want full control of an animation. Aside from adjusting the raw vertex motions predicted by the voice puppet, there are several ways an animator could intercede to customize the animation. Here we list a few, beginning with the easiest: (1) Choose from a palette of puppets, each trained on a different style of speech and facial mannerisms. (2) Increase the variance of the training data, which produces a cartoon-like exaggerated range of motion in facial expression. (3) Add whole-face expression vectors (e.g., a grin) to those generated by the voice puppet. (4) Edit the facial state sequence. Options 3&4 are analogous to the present-day practices of superimposing multiple morph targets and editing a phoneme sequence, respectively.

7 Summary

Voice puppetry combines the voice, face, and facial mannerisms of three different people into a realistic speaking animation. Given novel audio, the system accurately generates lip and whole-face motions in the style of the training performance, even reproducing subtle effects such as co-articulation. This purely data-driven approach stands on two innovations: An entropy-minimization algorithm learns extremely compact and accurate probabilistic models of the facial behavior manifold from training video; a closed-form solution for geodesics on this manifold yields facial motion sequences that are optimally compatible with new audio and with learned facial behavior.

8 Acknowledgments

Thanks to interns and code contributors: The HMM and geodesic code was optimized by Ken Shan. The face renderer [25] was graciously provided to us by Robert Forchheimer, and modified for real-time animation by Ilya Baran. The texture tracker [15] was obtained from Greg Hager and modified to include mesh constraints by Ken Shan and Jon Yedidia. Ilya Baran, Jane Maduram, and Ken Mathews performed for training data and helped to process and analyze it. Thanks to the Rickover Science Institute for providing these talented high school interns. The acoustic analysis code [16] was obtained from the Berkeley International Computer Science Institute web site. Finally, thanks to anonymous reviewers and to colleagues who previewed this work at the 1998 Workshop on Perceptual User Interfaces for their comments and questions.

References

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

[2] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[3] 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.

[4] M. Brand. Structure discovery in conditional probability models via an entropic prior and parameter extinction. *Neural Computation (accepted 8/98)*, October 1997.

[5] M. Brand. Pattern discovery via entropy minimization. In *Proc. Artificial Intelligence and Statistics #7*, Morgan Kaufmann Publishers. January 1999.

[6] M. Brand. Shadow puppetry. Submitted to *Int. Conf. on Computer Vision, ICCV '99*, 1999.

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

[8] T. Chen and R. Rao. Audio-visual interaction in nultimedia communication. In *Proc. ICASSP '97*, 1997.

Figure 7: The many faces of a voice puppet.

[9] 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*. Springer-Verlag, 1993.

[10] S. Curinga, F. Lavagetto, and F. Vignoli. Lip movement sythesis using time delay neural networks. In *Proc. EUSIPCO '96*, 1996.

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

[12] T. Ezzat and T. Poggio. MikeTalk: A talking facial display based on morphing visemes. In *Proc. Computer Animation Conference*, June 1998.

[13] G.D. Forney. The Viterbi algorithm. *Proc. IEEE*, 6:268–278, 1973.

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

[15] G. Hager and K. Toyama. The XVision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1) pp. 23–37. 1997.

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

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

[18] J.P. Lewis. Automated lip-sync: Background and techniques. *J. Visualization and Computer Animation*, 2:118–122, 1991.

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

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

[21] K. Stevens (MIT). Personal communication., 1998.

[22] S. Morishima and H. Harashima. A media conversion from speech to facial image for intelligent man-machine interface. *IEEE J. Selected Areas in Communications*, 4:594–599, 1991.

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

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

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

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

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

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

[29] E. Yamamoto, S. Nakamura, and K. Shikano. Lip movement synthesis from speech based on hidden Markov models. In *Proc. Int. Conf. on automatic face and gesture recognition, FG '98*, pages 154–159, Nara, Japan, 1998. IEEE Computer Society.