Audio/Visual Mapping With Cross-Modal Hidden Markov Models

Shengli Fu, *Student Member, IEEE*, Ricardo Gutierrez-Osuna, *Member, IEEE*, Anna Esposito, Praveen K. Kakumanu, *Student Member, IEEE*, and Oscar N. Garcia, *Life Fellow, IEEE*

***Abstract*—The audio/visual mapping problem of speech-driven facial animation has intrigued researchers for years. Recent research efforts have demonstrated that hidden Markov model (HMM) techniques, which have been applied successfully to the problem of speech recognition, could achieve a similar level of success in audio/visual mapping problems. A number of HMM-based methods have been proposed and shown to be effective by the respective designers, but it is yet unclear how these techniques compare to each other on a common test bed. In this paper, we quantitatively compare three recently proposed cross-modal HMM methods, namely the remapping HMM (R-HMM), the least-mean-squared HMM (LMS-HMM), and HMM inversion (HMMI). The objective of our comparison is not only to highlight the merits and demerits of different mapping designs, but also to study the optimality of the acoustic representation and HMM structure for the purpose of speech-driven facial animation. This paper presents a brief overview of these models, followed by an analysis of their mapping capabilities on a synthetic dataset. An empirical comparison on an experimental audio-visual dataset consisting of 75 TIMIT sentences is finally presented. Our results show that HMMI provides the best performance, both on synthetic and experimental audio-visual data.**

***Index Terms*—3-D audio/video processing, joint media and multimodal processing, speech reading and lip synchroization.**

I. Introduction

THE GOAL OF audio/visual (A/V) mapping is to produce accurate, synchronized and perceptually natural animations of facial movements driven by an incoming audio stream. Speech-driven facial animation can provide practical benefits in human-machine interfaces [1], since the combination of audio and visual information has been shown to enhance speech perception, especially when the auditory signals degrade due to noise, bandwidth filtering, or hearing impairments.

Despite its apparent simplicity, the mapping between continuous audio and visual streams is rather complex as a result of co-articulation [2], which causes a given phone to be pronounced differently depending on the surrounding phonemes. According to the level at which speech signals are represented, facial animation approaches can be classified into two groups: phoneme/viseme and subphonemic mappings. The phoneme/viseme approach views speech as a bimodal linguistic entity. The basic idealized linguistic unit of spoken language is the phoneme. The spoken English language has approximately 58 phonemes [3]. Similarly, the basic unit of facial speech movement corresponding to a phoneme is the viseme. Following Goldschen [4], phonemes may be mapped into 35 different visemes. Although phoneme/viseme mappings have generated fairly realistic “talking-heads” [5], [6] the approach is inherently limited. When speech is segmented into phonemes, considerable information is lost, including speech rate, emphasis and prosody, all of which are essential for animating a realistically perceived talking face. Therefore, phoneme/viseme mappings result in less natural facial animations.

An alternative to the phoneme/viseme approach is to construct a direct mapping from sub-phonemic speech acoustics (e.g., linear predictive coefficients) onto orofacial trajectories. This approach assumes a dynamic relationship between a short window of speech and the corresponding visual frame. In this case, the problem consists of finding an optimal functional approximation using a training set of A/V frames. As universal approximators, neural networks have been widely used for such nonlinear mapping [7]. To incorporate the co-articulation cues of speech, Lavagetto [8] and Massaro *et al.* [9] proposed a model based on time-delay neural networks (TDNNs), which uses tapped-delay connections to capture context information during phone and motion transitions. Hong *et al.* [10] used a family of 44 multilayer perceptrons (MLP), each trained on a specific phoneme. Co-articulation is captured with a seven-unit delay line. An incoming audio sample is first classified into a phoneme class using a set of 44 Gaussian mixture models, and the corresponding MLP is then used to predict the video components. Obviously, the predefined length of the delay line limits the time window of co-articulation dynamics because the context and durations are quite different for different subjects, emotional states, and speech rates.

To overcome the above limitations, hidden Markov models (HMMs) have recently received much attention for the purpose of A/V mapping [11]–[14]. HMM-based methods have the advantage that context information can be easily represented by state-transition probabilities. To the best of our knowledge, the first application of HMM to A/V mapping is the work by Yamamoto *et al.* [14]. In this approach, an HMM is learned from

Manuscript received March 12, 2003; revised October 5, 2003. This work was supported by the National Science Foundation (NSF) under Grant EIA-9906340, Grant BCS-9980054, and CAREER Award 9984426/0229598. The associate editor coordinating the review of this paper and approving it for publication was Prof. Suh-Yin Lee.

S. Fu is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: fu@ee.udel.edu).

R. Gutierrez-Osuna is with Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: rgutier@cs.tamu.edu).

A. Esposito is with the Department of Psychology, Second University of Naples, Naples, Italy (e-mail: iiass.anna@tin.it).

P. K. Kakumanu is with the Department of Computer Science and Engineering, Wright State University, Dayton, OH 45435 USA (e-mail: kpraveen@cs.wright.edu).

O. N. Garcia is with the College of Engineering, University of North Texas, Denton, TX 76203 USA (e-mail: garcia@unt.edu).

Digital Object Identifier 10.1109/TMM.2005.843341

audio training data, and each video training sequence is aligned with the audio sequence using Viterbi optimization. During synthesis, an HMM state sequence is selected for a given novel audio input using the Viterbi algorithm, and the visual output associated with each state in the audio sequence is retrieved. This technique, however, provides video predictions of limited quality for two reasons. First, the output of each state is the average of the Gaussian mixture components associated to that state, so the predicted visual output of this state is only indirectly related to the current audio vector by means of the Viterbi state. Second, synthesis performance depends heavily on the Viterbi alignment, which is rather sensitive to noise in the audio input.

Chen and Rao [11], [15] employ a least mean square estimation method for the synthesis phase, whereby the visual output is made dependent not only on the current state, but also on the current audio input. Still, their model predicts the state sequence by means of the Viterbi algorithm. To address this limitation, Choi *et al.* [12] have proposed a hidden Markov model inversion (HMMI) method. In HMMI, the visual output is generated directly from the given audio input and the trained HMM by means of an expectation-maximization (EM) iteration, thus avoiding the use of the Viterbi sequence. A different mechanism has been proposed by Brand [13], in which a minimum-entropy training method is used to learn a concise HMM. As a result, the Viterbi sequence captures a larger proportion of the total probability mass, thus reducing the detrimental effects of noise in the audio input. For reasons that will become clear in Section II-A, Brand's method is termed the remapping HMM (R-HMM).

Even though these three HMMs have shown high potential for A/V mapping in speech-driven facial animation systems, a clear understanding of their relative merits has yet to be discerned. Moreover, the proposed models have not yet been evaluated under the same experimental conditions or A/V datasets, but the originators have used their own datasets. A theoretical evaluation alone may miss important factors affecting their behavior since the performance of any computational model will ultimately depend on the problem domain and the nature of the data. The objective of this paper is to provide an experimental comparison of these HMMs, as well as investigate how HMM structure and choice of acoustic features affect the prediction performance for speech-driven facial animation.

II. HMM-Based A/V Mapping

An HMM is commonly represented by a vector of model parameters $\lambda = (S, O, A, B, \pi)$, where $S = (s_1, s_2, \ldots, s_N)$ is the set of Markov chain states, O denotes the set of observations, A is a matrix of state transition probabilities, and π is the initial state distribution. If the outputs of the HMM are discrete symbols, B is the observation symbol probability distribution. Although continuous outputs can be discretized through vector quantization, improved performance can be obtained by modeling the output probability distribution at state j $B = \{b_j(O)\}$ with a semi-parametric Gaussian mixture model

$$b_j(O) = \sum_{m=1}^{M} c_{jm} N(O; \mu_{jm}, U_{jm}) \tag{1}$$

where c_{jm} is the mixture coefficient for the mth mixture at state j, and $N(\cdot)$ is a Gaussian density with mean vector μ_{jm} and covariance matrix U_{jm} [16]. As reviewed in the previous section, a number of extensions of this basic HMM have been proposed for A/V mappings. A concise description of these models follows.

A. Remapping HMM

Under the assumption that both acoustic and visual data can be modeled with the same structure, Brand [13] has proposed a remapping procedure to train cross-modal HMMs. The training is conducted using video data. Once a video HMM is learned, the video output probabilities at each state are remapped onto the audio space using the M-step in Baum-Welch. Borrowing notation from [16], the estimation formulas for μ_{jm} and U_{jm}, the mean and covariance for the mth Gaussian component at the jth state in the visual HMM are

$$\begin{aligned} \mu_{jm} &= \frac{\sum_{t=1}^{T} \gamma_t(j,m) v_t}{\sum_{t=1}^{T} \gamma_t(j,m)} \\ U_{jm} &= \frac{\sum_{t=1}^{T} \gamma_t(j,m)(v_t - \mu_{jm})(v_t - \mu_{jm})^T}{\sum_{t=1}^{T} \gamma_t(j,m)} \end{aligned} \tag{2}$$

where v_t is the visual vector at time t, and $\gamma_t(j,m) = P(q_t = S_j, g = G_m | V, \lambda)$ is the probability of being in state S_j at time t with the mth mixture component accounting for visual sequence V and learned model λ. To re-map the video HMM into audio space, the audio μ_{jm} and U_{jm} are obtained by replacing the video vector v_t in (2) with a_t, the audio vector at time t. All other parameters in the audio HMM remain the same as in the video HMM.

The process of synthesizing a novel video trajectory sequence involves two steps. First, given a new audio sequence and the learned audio HMM, the optimal state sequence is obtained with the Viterbi algorithm. From the Viterbi sequence, the A/V mapping may be simply implemented by choosing the average visual vector for each state, as in [14]. This naive solution, however, yields an animation which displays jerky motion from frame to frame. Instead, Brand [17] proposes a solution that yields a short, smooth trajectory that is most consistent with the visual HMM and the given Viterbi state sequence. For simplicity, each state is assumed to have one Gaussian component, but the procedure can be generalized to Gaussian mixtures. Let $N(O; \mu, U)$ be the probability of observation O given Gaussian model with mean μ and covariance U. The predicted visual trajectory O^* is then

$$\begin{aligned} O^* &= \arg\max_{O} \log \prod_{t} N(o_t; \mu_{s(t)}, U_{s(t)}) \\ &= \arg\min_{O} \sum_{t=1}^{T} (o_t - \mu_{s(t)})^T U_{s(t)}^{-1} (o_t - \mu_{s(t)}) \end{aligned} \tag{3}$$

where $o_t = [v_t\ \dot{v}_t]^T$, $\mu_{s(t)} = [\mu_{s(t)}^{v}\ \mu_{s(t)}^{\dot{v}}]^T$. Thus, the observation o_t at time t for the visual HMM includes both the position v_t

and the velocity $\dot{v}_t$. Equation (3) has a closed-form solution with a single global optimum. A standard block tri-diagonal system can be obtained by setting its derivative to zero. Details of the solution for such system can be found in [18] and [19].

B. Least-Mean Squared HMM

The LMS-HMM method of Chen [11] differs from the R-HMM method in two fundamental ways. First, the LMS-HMM is trained on the joint A/V space, as opposed to video space. Second, the synthesis of video for each particular state is formulated as a least-mean-squares regression from the corresponding audio observation. Training of the LMS-HMM [11] is performed by combining the audio and visual features into one joint observation vector $o_t = [a_t \; v_t]^T$. Once the joint HMM is trained using Baum-Welch, the extraction of an audio HMM is trivial since the audio parameters are part of the joint A/V distribution

$$\mu_{jm} = [\mu_{jm}^{a} \; \mu_{jm}^{v}]^T; \quad U_{jm} = \begin{bmatrix} U_{jm}^{aa} & U_{jm}^{av} \\ U_{jm}^{va} & U_{jm}^{vv} \end{bmatrix} \tag{4}$$

where μ_{jm}^{a} and U_{jm}^{aa} represent the mean vector and covariance matrix for the mth Gussian component at the jth state in the audio HMM. To synthesize a video vector from a new audio input, the LMS-HMM method operates in two stages. First, the most likely state sequence is found based on the learned audio HMM using the Viterbi algorithm. Then, the audio input a_t and the Gaussian mixture model corresponding to each Viterbi state q_t are used to analytically derive the visual estimate $\hat{v}_t$ that minimizes the mean squared error (MSE) $E[(v_t - \hat{v}_t)^2 | a_t]$. It can be shown [15] that this MSE estimate is given by

$$\hat{v}_t = E[v_t | a_t] = \sum_i \left(\frac{c_i N(a_t; \mu_{a,i}, U_{a,i})}{f_{q_t}(a_t)} \right) \left(b_i^T \begin{bmatrix} 1 \\ a_t \end{bmatrix} \right) \tag{5}$$

with

$$\begin{aligned} b_i &= \begin{bmatrix} 1 & \mu_{a,i}^T \\ \mu_{a,i} & U_{aa,i} \end{bmatrix}^{-1} \begin{bmatrix} \mu_{v,i} \\ U_{av,i} \end{bmatrix} \\ f_{q_t}(a_t) &= \sum_i c_i N(a_t; \mu_{a,i}, U_{a,i}) \end{aligned} \tag{6}$$

where c_i is the mixture coefficient, and $N(a_t; \mu_{a,i}, U_{a,i})$ is the probability of a_t for the ith Gaussian component in state q_t. Equation (5) shows that the visual output for a given state depends directly on the corresponding audio input.

C. HMM Inversion

The HMMI approach of Choi *et al.* [12], [20] addresses a major weakness of HMMs: reliance on the Viterbi sequence, which represents only a small fraction of the total probability mass, with many other state sequences potentially having nearly equal likelihoods [13]. In addition, the Viterbi search may be easily misguided by noise in the audio input. To avoid these problems, in HMMI the visual outputs are estimated directly from the speech signal, bypassing the Viterbi search [21].

The training process for HMMI is the same as in the LMS-HMM method, where both audio and video features are used to train a joint A/V HMM. During synthesis, the visual output in HMMI is predicted from the given audio stimuli and the joint HMM using a procedure that can be regarded as the inverse version of Baum-Welch. The objective of the HMMI re-estimation method is to find a video observation sequence $\hat{v}$ that maximizes Baum's auxiliary function [22]

$$Q\left(\lambda_{av}, \lambda_{av}; a, v, \hat{v}\right) \tag{7}$$

where a is the audio sequence, v is an initial visual sequence, and λ_{av} are the parameters of the joint A/V HMM. Note that (7) has two identical λ_{av} since the EM step in HMMI does not re-estimate model parameters but the video observation sequence $\hat{v}$. Details of this derivation may be found in [12]. Since both HMMI and LMS-HMM are trained on the joint A/V space, our HMMI implementation uses the visual sequence predicted by LMS-HMM as an initial value v in (7).

III. Evaluation on Synthetic Data

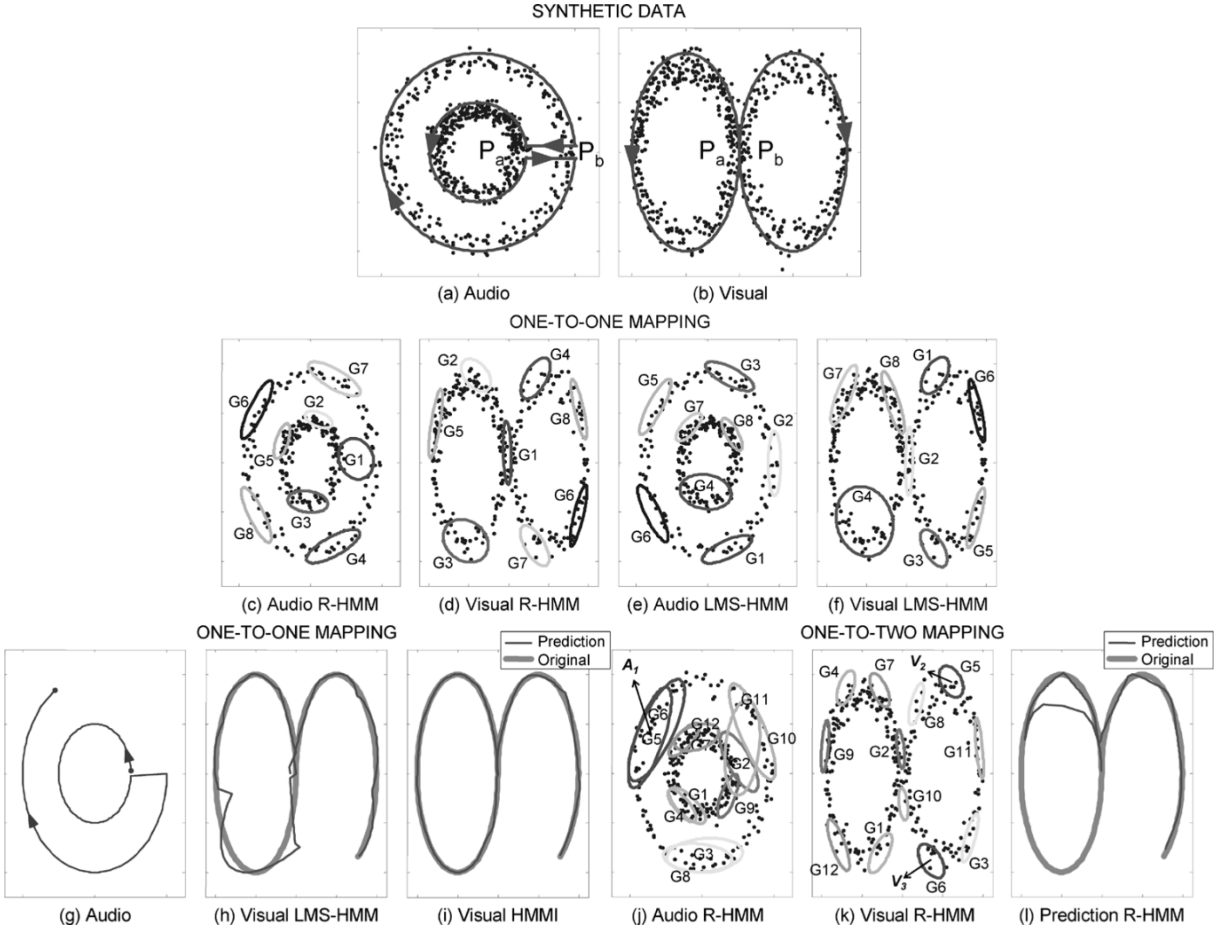
Section II has presented a brief description of three HMM-based A/V prediction methods. For a better understanding of their capabilities, we first present an empirical evaluation on a synthetic dataset adapted from [23]. Illustrated in Fig. 1(a) and (b), this dataset simulates audio and video by two-dimensional feature vectors following circular trajectories.

- The motion of the synthetic "audio" vector is along two concentric circles. The audio signal moves counterclockwise along the inner circle or clockwise along the outer circle. Transitions from one circle to the other can only occur at points P_a and P_b.
- The synthetic "visual" signal moves along an eight-shaped trajectory, but the movement is synchronized with the audio input. When the audio signal is traversing the inner circle, the visual signal moves counterclockwise along the left ellipse. Otherwise, the visual signal will move clockwise along the right ellipse.

The angular velocity of the audio and visual signals may be different. Three cases are considered. *First*, when audio and video have the same speed, the resulting mapping is one-to-one: only one visual position corresponds to the given audio position, and vice versa. *Second*, if the visual signal moves at a higher velocity than the audio signal, more than one audio position will correspond to the same visual position. This situation represents a many-to-one mapping. *Finally*, if the audio signal moves with a higher velocity than the visual signal, the mapping is one-to-many: one audio vector may have different visual counterparts. In the remaining part of this section, we evaluate the synthesis performance of the three HMM algorithms on these different mappings. For simplicity, a two-to-one and a one-to-two mapping will serve as particular cases of the many-to-one and one-to-many mappings.

A. One-to-One and Two-to-One Mappings

For the one-to-one and two-to-one mapping, the three HMMs successfully capture the inherent relation between the audio and visual spaces. Here we only present the one-to-one mapping



(a) Audio (b) Visual

(c) Audio R-HMM (d) Visual R-HMM (e) Audio LMS-HMM (f) Visual LMS-HMM

(g) Audio (h) Visual LMS-HMM (i) Visual HMMI (j) Audio R-HMM (k) Visual R-HMM (l) Prediction R-HMM

Fig. 1. (a), (b) Synthetic A/V dataset (adapted from [23]). Learned models for one-to-one mapping using (c), (d) R-HMM and (e), (f) LMS-HMM. Synthesized trajectory for one-to-one mapping: (g) incoming audio sequence, (h) visual outputs for LMS-HMM, and (i) HMMI. Learned model for one-to-two mapping with (j), (k) R-HMM and (l) predicted trajectory.

result; the two-to-one mapping displays similar properties. When the audio and visual vectors have the same velocity, each point in the audio domain maps to a unique point in the visual domain. Fig. 1(d) shows the Gaussian mixture of the video data obtained with R-HMM. Using the re-mapping (2), the mixture shown in Fig. 1(c) is obtained for the audio data. In this case, an eight-state continuous HMM was used, each state having one Gaussian component to facilitate visualization of the A/V mappings. The distribution of Gaussian components in the audio HMM not only captures the spatial structure of the synthetic audio data, but also models their dynamics. From the visual HMM, the sequences of Gaussian components $\{G_1G_2G_5G_3\}$ and $\{G_1G_4G_8G_6G_7\}$ can be observed for the left and right ellipses, respectively. These are the same sequences for the audio signals, rotating counterclockwise for the inner circle and clockwise for the outer circle. Results for the LMS-HMM and HMMI models, shown in Fig. 1(e) and (f), are qualitatively similar. In these two cases, however, the audio and visual Gaussian mixtures are extracted directly from the joint HMM using (4). The HMMI improvements are illustrated in Fig. 1(g)–(i) for the one-to-one case. Fig. 1(g) represents the audio stimulus. Fig. 1(i) shows the HMMI prediction, a smoother and more accurate visual trajectory than the LMS-HMM prediction in Fig. 1(h), which served as the initial estimate.

B. One-to-Two Mapping

In order to investigate one-to-many mappings, the velocity of the audio will now be set to twice that of the visual trajectory. This corresponds to a situation where a single audio vector maps to two positions in video space. Fig. 1(j) and (k) shows the structure for the audio and video data using R-HMM. In this case, an audio sample A_1 can be mapped to two different visual vectors: V_2 and V_3. Although the re-mapping accurately captures the A/V relationship, the Viterbi algorithm is unable to provide the correct state sequence for the visual data since the Gaussian audio components for different video vectors are overlapping (e.g., G_5 and G_6, G_4 and G_1). Even though entropic learning may generate a more concise structure during the training of a visual HMM [24], a succinct audio HMM cannot be guaranteed to produce a correct solution since it is obtained through a re-mapping process. If different video components share the same audio component, the overlap is unavoidable. Fig. 1(l) shows the synthesized trajectories, which further illustrate this problem.

A similar argument can be used to explain why the LMS-HMM method (not shown here) also fails on the one-to-many case. Although this model is trained using both audio and visual data, the one-to-many mapping will inherently generate more than one Gaussian component for the same point, which easily leads to the incorrect Viterbi sequence. In the case of HMMI, the Viterbi search is bypassed. However, since the initial guess is the output of the LMS-HMM technique, HMMI can only provide a local search around this initial solution.

Unsatisfactory results are obtained for the one-to-two mapping, since none of the three HMMs can capture enough context information to preserve the history of the trajectory. However, since the relationship between phones and visemes can be described as being at worst three-to-one [4], and never one-to-many, the three models show potential for real A/V mapping problems. This is because we expect that, taking into account the context of the frames that constitute these phone-to-viseme mappings, reality will be unlikely to exhibit one-to-many mappings. This issue is explored in the following sections, where the prediction performance is validated on real data from a speech-driven facial animation system.

IV. Speech-Driven Facial Animation System

A baseline A/V capture and speech-driven facial animation system has been developed at Wright State University during the past few years. Our methodology differs from other approaches in several notable features. First, the capture system samples video at 60 frames per second (fps), as compared to most other facial animation systems, which operate at 30 fps and miss important visual details of articulation [25]. Second, our preprocessing module generates a rich set of acoustic features by combining information from five different speech preprocessing methods. Earlier work by Kakumanu [26] showed that the combination of acoustic features provides better prediction results than those obtained using features from a single preprocessing method. Third, as compared to the phoneme/viseme mapping, which results in a significant loss of prosodic information, our system operates at the sub-phonemic level, thus becoming language-independent. Lastly, the visual parameters employed in the system follow the MPEG-4 standard, allowing for an easy and direct use in facial-animation applications.

A. Audio Processing

The purpose of audio processing is to extract robust speech features that are predictive of orofacial motion. Unfortunately, not much is known about which acoustic features are relevant for speech-driven facial animation. Therefore, three general types of features, each providing a different characterization of the speech signal, are selected for evaluation in our system.

- *Prosodic:* fundamental frequency (F0) and signal energy.
- *Articulatory:* Linear predictive coefficients (LPCs) and linear spectral frequencies (LSFs) [27].
- *Perceptual:* Mel frequency cepstral coefficients (MFCCs) [27] and perceptual critical band features (PCBFs) [28].

In addition, we also consider a hybrid representation (COMB), which combines the individual speech feature vectors into a large hybrid vector. A compressed hybrid representation is also considered by projecting this COMB feature vector along the Fisher's linear discriminant (LDA) projection [29], which provides the audio subspace that best discriminates the various facial configurations. Our experience shows that the resulting LDA projection can be truncated with ten eigenvectors (or 90% of the variance) without a significant loss of information [26].

B. Video Processing

To facilitate accurate tracking of the facial dynamics, 27 markers are placed on the subject's face at various positions defined by the MPEG-4 standard [30]. The three-dimensional (3-D) coordinates of these markers are then recovered through stereo vision [31]. This process yields a video vector with 81 (27×3) measurements, which are highly correlated since movements of the points on the face are highly interdependent. For this reason, principal component analysis (PCA) is used to project this redundant video data onto a low-dimensional space that preserves most of the relevant motion. Based on previous results [31], we use four PCs as the reduced video vector for the HMM training. The 3-D position of the 27 original markers is then recovered from these four PCs through least-squared back-projection. Prediction performance of the HMMs is evaluated in terms of the three representative orofacial articulators: mouth height $(MH = 8.1.y - 8.2.y)$, mouth width $(MW = 8.3.x - 8.4.x)$, and chin height $(CH = 9.3.y - 2.10.y)$, which are obtained from the 3-D positions of MPEG-4 facial points.

C. Audio/Visual Dataset

An A/V database of 75 sentences from the TIMIT Speech Corpus [32] compact set was collected from a single speaker to evaluate the system. Each sentence was repeated five times, for a total of 375 sentences. The 75×5 sentences were split into three separate sets, a training set containing 60×4 sentences, a validation set containing the fifth repetition from the previous 60 sentences, and a test set containing one repetition from the remaining 15×5 sentences. The use of a validation and test set allows us to identify an upper and lower bound of predictive accuracy, respectively.

V. Experimental Results

This section presents a quantitative comparison of the three HMM algorithms in terms of their ability to predict real orofacial motion. The effects of context information and HMM structure are also investigated. Prediction performance will be measured using the normalized MSE ε and correlation coefficient (CC) ρ between the predictions $\hat{v}_t$ (MH, MW, and CH) and the true trajectories v_t

$$\varepsilon = \frac{1}{\sigma_v^2}\frac{1}{T}\sum_{t=1}^{T}(\hat{v}_t - v_t)^2; \quad \rho = \frac{1}{T}\sum_{t=1}^{T}\frac{(v_t - \mu_v)(\hat{v}_t - \mu_{\hat{v}})}{\sigma_v \sigma_{\hat{v}}}. \tag{8}$$

A. Effect of Audio Context

Speech is the result of a dynamic articulation process moving through time, and the resulting facial motion is affected not only by the movements required to produce a speech segment at a given instant but also by the movements required to actuate the transition from the preceding (backward co-articulation) to the

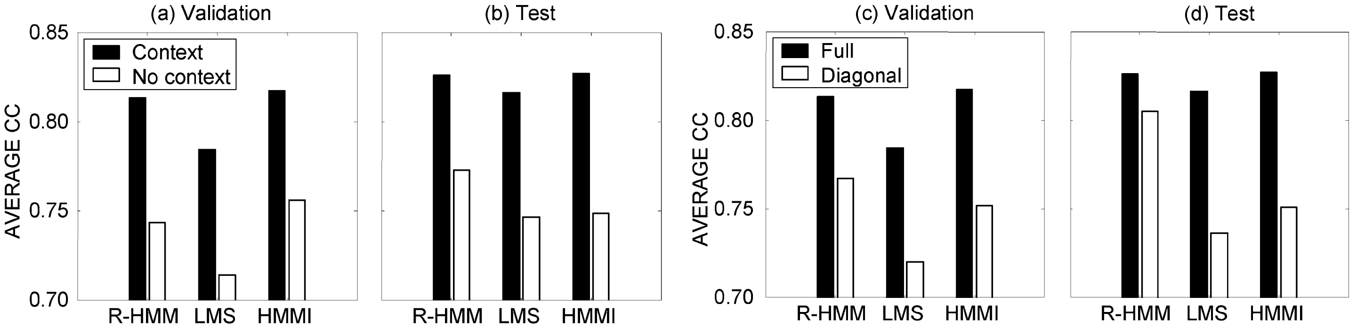


Fig. 2. Effects of (a), (b) context and (c), (d) covariance structure.

upcoming (forward co-articulation) speech segment. Therefore, this context should be taken into account if orofacial motion is to be predicted from audio. To encode context in the audio representation, we associate each video frame $v(k)$ with the acoustic features $a(k)$ of the same frame index k, as well as past and future frames by means of a tapped-delay line. Our prior work [26] has shown that a context length of five frames (assuming video capture at 60 fps) is appropriate to account for backward and forward co-articulatory effects

$$A(k) = [a(k-5), a(k), a(k+5)] \tag{9}$$

where $A(k)$ is the acoustic feature vector carrying context information. The final result of this process is a database of A/V dynamic pairs $[A, v](k)$, which constitutes the training data. To illustrate the relevance of context, two separate HMMs were trained, one using the context-rich pairs $[A, v](k)$, and a second one using a context-less (frame-to-frame) pair $[a, v](k)$. Audio data was processed with the LDA technique described in Section IV-A. Qualitatively similar results were also obtained with other audio preprocessing techniques and are, therefore, not reported here. The results are shown in Fig. 2(a)–(b), where the average correlation coefficient for the three orofacial parameters (MH, MW, and CH) predicted by the three HMMs is reported for both validation and test data. These results clearly indicate that context information heavily contributes to the visual prediction, with a 5% to 9% improvement over the context-less features. Therefore, the remaining experiments will consistently use audio features with context information as described by (9).

B. Effect of Covariance Structure

Continuous HMMs describe the observation outputs of each state with a Gaussian Mixture ($1 \leq \mathrm{m} \leq \mathrm{M}$) parameterized by μ_{jm} and U_{jm}, the mean vector and covariance matrix of the mth Gaussian component at state j. The covariance matrix for each Gaussian component can be either diagonal, which assumes that the features are statistically independent, or full, which also considers dependencies among features. To determine the appropriate covariance structure, the three HMM algorithms were trained using both diagonal and full covariance matrices. Results are shown in Fig. 2(c)–(d) for HMMs with 24 states (one Gaussian component per state), in terms of the average correlation coefficient of the three orofacial parameters. Audio data was processed with the LDA technique. Qualitatively similar results were obtained with other audio preprocessing techniques. These results clearly show that the use of a full covariance matrix significantly increases the prediction performance, with an average increase of 7%. Consequently, a full covariance matrix is used for the remaining experiments presented in this paper.

C. Effect of Audio Preprocessing Choice

The various speech-preprocessing techniques (e.g., prosodic, articulatory, perceptual) capture unique pieces of information that are useful for predicting orofacial motion [26]. Therefore, it is worthwhile to investigate the performance of the HMMs on different speech features. This section presents an experimental comparison of five individual speech preprocessing choices: 1) energy and F0 (termed PP in what follows); 2) LPC; 3) LSF; 4) MFCC; and 5) PCBF. The PP representation consists of a six-dimensional vector (2 features/frame × 3 frames), whereas each of the remaining four techniques consists of a 36-dimensional vector (12 coefficients/frame × 3 frames). As mentioned in Section IV-A, we also consider the hybrid representation COMB, which yields a 150-dimensional hybrid vector ($36 \times 4 + 6$), and its ten-dimensional LDA projection.

Fig. 3(a) and (b) shows the prediction performance of these seven preprocessing techniques in terms of the average correlation coefficients on the three orofacial parameters for the HMMI model. The performance of R-HMM and LMS-HMM is qualitatively similar to that of HMMI, so their results are not included. It can be noted that COMB, LDA, MFCC, LPC, LSF and PCBF present similar performance on validation data, followed by PP. On test data, COMB and LDA produce the highest performance, followed by LSF, MFCC, PCBF and LPC. PP consistently provides the lowest predictive accuracy, a reasonable result since energy and "pitch" information can only partially explain the motion of orofacial articulators. Although the 150-dimensional COMB audio vector captures a wealth of information in the audio signal, it comes at the expense of lengthy training sessions. As a result, the HMMI/COMB model with 48 states is unable to reach convergence after 500 iterations, or about one week of CPU time on a 2.4-GHz Pentium IV PC, which explains the degraded performance when going from 36 to 48 states in Fig. 3(b). This is also the reason why results with 64 states are not presented for the HMMI/COMB model. Considering these results, one can draw the conclusion that COMB and LDA are capable of extracting information that generalizes well for phonetic sequences not included in the training set. Although COMB provides the best results when HMMI reaches convergence, the LDA transformation results in a very close performer

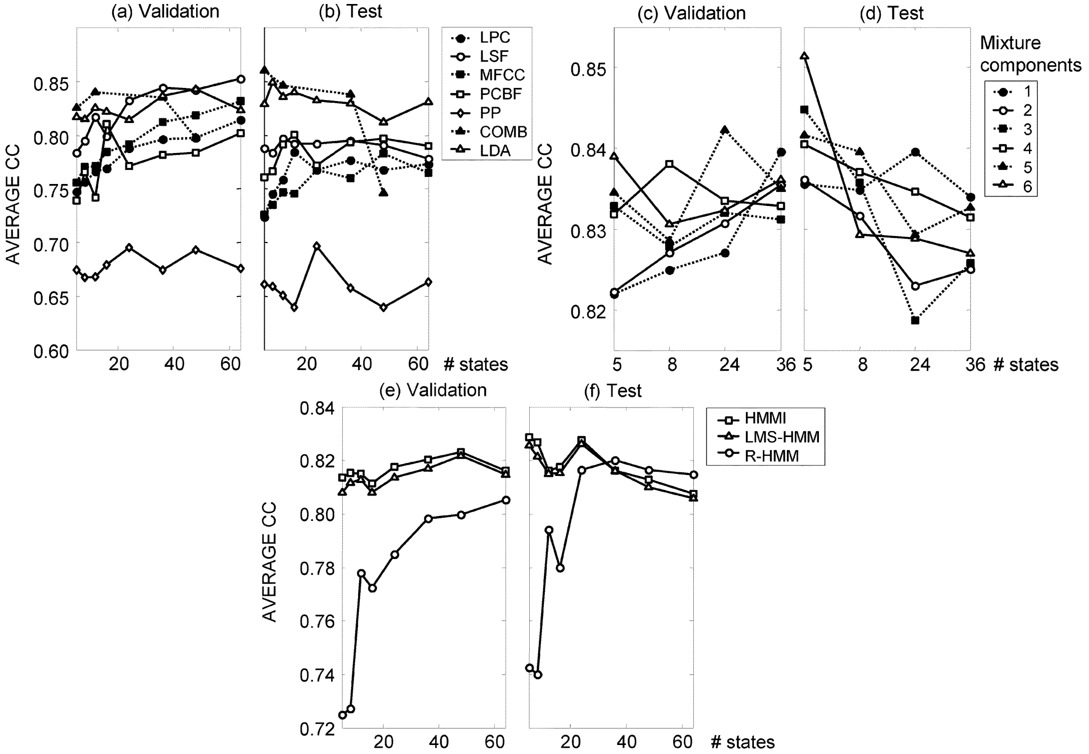


Fig. 3. (a), (b) HMMI performance versus audio representation and (c), (d) number of mixtures per state. (e), (f) Performance of the three HMMs.

with a more compact representation (a 15:1 reduction in dimensionality). For this reason, the next sections will focus the discussion on HMMs trained with LDA features.

D. Effect of Number of States and Number of Mixture Components per State

Fig. 3(a) and (b) also demonstrates the effects of varying the number of HMM states from 5 to 64. To isolate the effect of the number of states, the number of mixture components in each state was fixed to one. It can be observed that the prediction performance on validation data improves with the number of HMM states. The performance on test data, however, stabilizes around $\mathrm{N} = 36$, indicating the point at which the HMM has enough parameters to start over fitting the phonetic sequences in the training (or validation) set. More importantly, considering the fact that visual speech of the English language contains 35 visemes [4], our experimental results justify a rationale for estimating the number of states. The number of states could be argued to be approximately equal to the number of visemes in the language, if possible (that is, if enough data is available).

To analyze the effects of the number of mixtures per state, a second experiment was performed using the HMMI model. The R-HMM and LMS-HMM show similar behavior and their results are not included. A set of HMMs was trained with $\mathrm{N} = 5$, 8, 24, and 36 states, and with $\mathrm{M} = 1$, 2, 3, 4, 5, and 6 mixtures per state. In each case, the LDA technique was used to encode audio information. For ease of visualization, only the average CC of the three orofacial parameters is reported. The results are shown in Fig. 3(c) and (d) for validation and test data. It can be seen that the CC is largely insensitive to the choice of M. The largest difference between the best and worst performance is less than 0.03, and occurs for test data when the model has 24 states. The fact that increasing the number of mixtures per state does not result in a consistent improvement in prediction performance is an indication that one or two mixture components per state may be adequate, at least for our speech-driven facial animation system.

E. Effect of Cross-Modal HMM Choice

The previous sections focused on the sensitivity of the A/V mapping to HMM structural parameters such as the form of the covariance matrix, and the number of states and mixtures. In this section, we will present a final comparison among the three HMMs. Based on the previous results, one Gaussian per state with a full covariance matrix will be used for all the HMMs, but the number of HMM states will remain as a free parameter. Fig. 3(e) and (f) illustrates the performance of the R-HMM, LMS-HMM and HMMI in terms of the average correlation coefficient on validation and test data for models with 5, 8, 12, 16, 24, 36, 48, and 64 states. LDA features are used in this comparison; results with the other audio features are qualitatively similar (although obviously lower). The results clearly indicate the superior performance of HMMI and LMS-HMM when compared with R-HMM. Although R-HMM presents a higher performance on test data set when the number of states is greater than 36, LMS-HMM and HMMI achieve higher performance with fewer states. Comparing the performance on validation and test data, it appears that the three models suffer from over-fitting for more than 36 states.

The difference in performance between HMMI and LMS-HMM, which share the same underlying joint HMM, can be justified by the fact that HMMI does not rely on the Viterbi search.

TABLE I
SUMMARY OF MODEL PERFORMANCE IN TERMS OF CC

			LPC	LSF	MFCC	PCBF	PP	COMB	LDA
R-HMM	VAL	MH	0.81	0.86	**0.87**	**0.87**	0.73	0.84	**0.87**
		MW	0.60	0.69	0.71	0.71	0.42	0.67	**0.72**
		CH	0.76	0.82	**0.84**	0.83	0.67	0.79	**0.84**
	TST	MH	0.80	0.85	0.85	**0.86**	0.69	0.84	**0.86**
		MW	0.65	0.69	0.68	0.65	0.42	0.69	**0.78**
		CH	0.76	0.82	0.80	0.83	0.65	0.82	**0.84**
LMS-HMM	VAL	MH	0.85	**0.89**	0.87	0.86	0.82	0.88	0.88
		MW	0.71	0.75	0.73	0.70	0.49	0.75	**0.76**
		CH	0.82	**0.86**	0.85	0.83	0.75	0.85	0.84
	TST	MH	0.81	0.84	0.84	0.85	0.81	**0.88**	0.87
		MW	0.69	0.71	0.63	0.67	0.44	**0.76**	**0.76**
		CH	0.77	0.82	0.79	0.83	0.75	**0.87**	0.86
HMMI	VAL	MH	0.87	**0.88**	0.87	**0.88**	0.83	**0.88**	**0.88**
		MW	0.72	0.74	0.73	0.70	0.49	0.75	**0.76**
		CH	0.84	**0.85**	0.84	0.83	0.75	**0.85**	**0.85**
	TST	MH	0.84	0.84	0.84	0.87	0.82	**0.88**	0.87
		MW	0.71	0.72	0.62	0.66	0.45	**0.76**	**0.76**
		CH	0.82	0.83	0.80	0.84	0.75	**0.87**	0.86

TABLE II
RUN TIME (IN SECONDS) FOR THE THREE HMMS

MODEL SIZE		TRAINING			SYNTHESIS					
N M=1	M N=12	LMS-HMM			RHMM		LMS-HMM		HMMI*	
		N		M	N		N		N	
		LDA	LPC	LDA	LDA	LPC	LDA	LPC	LDA	LPC
5	1	59	86	73	2.6	2.9	1.8	11	1.2	2.4
12	2	74	162	93	2.7	2.9	2.1	12	1.6	4.6
24	4	97	278	140	2.7	2.9	2.0	13	3.1	9.2
48	8	240	648	246	2.9	2.9	2.2	14	5.8	17.6

**Runtimes are for the final EM fine-tuning only.*

Instead, HMMI generates the maximum likelihood visual sequence directly from the joint HMM and the initial solution provided by LMS-HMM. The additional EM stage in HMMI consistently increases the prediction performance, both on training and test data. This fine-tuning is, however, limited to solutions near the LMS-HMM solution due to the fact that EM is a local search. In our experiments, EM converges after a few iterations (typically less than 10), which shows that HMMI can be used as a final fine-tuning step at a small computational cost.

F. Summary

Table I summarizes the performance of the three A/V mapping techniques for each of the seven audio features (LPC, LSF, MFCC, PCBF, PP, COMB and LDA) on validation and test data. The best performer for each lip articulatory parameter is highlighted with underlined bold fonts. Regardless of the acoustic representation and the mapping method, these data indicates that the mouth width is the most difficult articulator to predict. For the three HMMs, COMB and LDA produce better performance than the other five acoustic vectors, supporting our hypothesis that different audio processing techniques encode for different visual information and that, therefore, a combined feature vector gives better results than the individual feature vectors alone. The results in Table I also show that the ten-dimensional LDA vector performs similar to the 150-dimensional COMB feature vector and, in the case of R-HMM, clearly better. Moreover, the most significant improvements of LDA over COMB are found in the mouth width, which is the hardest articulatory parameter to predict. These results show that LDA is an effective procedure to derive a linear projection of audio features that maximizes the discrimination of visual orofacial configurations.

A final but relevant element of comparison is the computational load of the three models. During training, R-HMM has a computational complexity of $O(TN^2 + TNMK^2)$, while LMS-HMM and HMMI are $O(TN^2 + TNM(D+K)^2)$, where T is length of the training sequence, N is the number of HMM states, M is number of mixtures per state, D is the dimensionality of the audio, and K is the dimensionality of the video. During synthesis, RHMM runs in $O(TN^2 + TNMD^2 + TK^3)$ and LMS-HMM is $O(TN^2 + TNMD^2 + TMKD)$. The synthesis phase of HMMI has the same computational complexity as the training phase, plus the cost of obtaining the initial solution with LMS-HMM. Table II shows the experimental runtimes of the models for different parameter settings, computed as the average of three training/synthesis sessions (60 × 4 training sentences, 15 test sentences) on a high-end PC workstation. The three models had very similar training times, so only those for LMS-HMM are provided. It is important to note that the audio LDA dimensionality-reduction step not only provides improved prediction performance but also significant time savings during the training and synthesis phases.

An example of the predicted (thin red line) and original trajectories (thick blue line) for each of the three orofacial articulators is shown in Fig. 4 for the TIMIT sentences: 1) *"Pizze-*

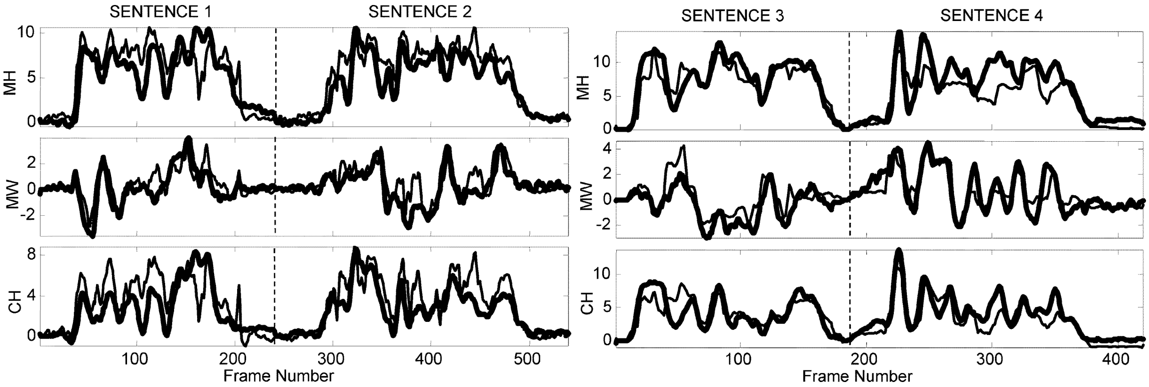


Fig. 4. Predicted (thin red line) versus actual (thick blue line) trajectories for HMMI.

rias are convenient for a quick lunch"; 2) *"The bungalow was pleasantly situated near the shore"*; 3) *"Clear pronunciation is appreciated"*; and (4) *"Barb's gold bracelet was a graduation present."* The first two sentences are from the validation set, whereas the last two sentences are from the test set. The predicted trajectories were passed through a mean filter (three frames wide) to remove jitter. These results were obtained with the HMMI technique and the LDA audio vector.

VI. Conclusions and Future Work

The following general conclusions regarding the various acoustic representations and HMM mapping techniques can be extracted from the empirical studies presented in this paper.

- Context information directly encoded into the acoustic representation can increase the prediction performance of the A/V mapping. Although HMMs have the modeling power to learn context information, explicitly incorporating acoustic context in the form of tap-delays results in a consistent improvement of predictive accuracy.
- Different acoustic representations carry complementary information regarding orofacial motion. The use of a combined hybrid vector provides improved performance as compared to the individual acoustic vectors. Moreover, the use of an LDA projection results in a 15:1 reduction in dimensionality with similar or better performance than the combined feature vector.
- Considering the structure of the HMM, our experiments suggest that the optimum model size lies between 24 and 48 states, with each state having 1 or 2 Gaussian components. Our results also show that a full covariance matrix for the Gaussian component results in an average 7% increase in performance when compared to a diagonal structure. It is interesting to note that, although these HMMs operate at a sub-phonemic level, the optimum number of states is roughly similar to the number of visemes.
- Among the three approaches, HMMI clearly provides the best performance, both on synthetic and experimental data. Although the HMMI solution is a local improvement of the initial LMS-HMM solution, its rapid convergence suggests it could be used as a final fine-tuning step at a small computational cost.

Generalization of our experimental results is limited by the restrictions on the A/V data examined. Presently, our dataset contains data from one American-born English speaker. In particular, noisy environments, multi-speaker and multi-language settings have not been considered. Expansion of the database along any one of these directions introduces additional representation problems and may result in different HMM structures. These questions are left for future studies. The principal conclusion of the study is that HMM-based A/V mappings are effective in capturing the information between acoustic and visual signals. The results do encourage a continuing effort to optimize the prediction performance by critical evaluation of each of the constituent components.

Acknowledgment

We wish to thank N. Balan, L. Moore and K. Das for their earlier contributions to the speech-driven facial animation system. We would also like to thank I. Rudomin, K. Murphy, and I. Nabney for their help in development and experimentation. K. Choi is also acknowledged for helpful discussions on the HMMI algorithm.

References

[1] I. Pandzic, J. Ostermann, and D. Millen, "User evaluation: Synthetic talking faces for interactive services," *Vis. Comput.*, vol. 15, no. 7–8, pp. 330–340, 1999.

[2] M. M. Cohen, J. Beskow, and D. W. Massaro, "Recent developments in facial animation: An inside view," in *Proc. AVSP*, Terrigal, Australia, 1998, pp. 201–206.

[3] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition* Englewood Cliffs, NJ, 1993.

[4] A. J. Goldschen, "Continuous Automatic Speech Recognition by Lipreading," Ph.D. dissertation, Eng. and App. Sci. Dept., George Washington Univ., Washington, NJ, 1993.

[5] S. Morishima, "Real-time talking head driven by voice and its application to communication and entertainment," in *Proc. AVSP*, Terrigal, Australia, 1998, pp. 195–199.

[6] C. Bregler, T. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Proc. ACM SIGGRAPH'97*, 1997, pp. 353–360.

[7] S. Morishima and H. Harashima, "A media conversion from speech to facial image for intelligent man-machine interface," *IEEE J Select. Areas Commun.*, vol. 9, no. 4, pp. 594–600, May 1991.

[8] F. Lavagetto, "Converting speech into lip movements: A multimedia telephone for hard of hearing people," *IEEE Trans. Rehabil. Eng.*, vol. 3, no. 1, pp. 90–102, Mar. 1995.

[9] D. W. Massaro, J. Beskow, M. M. Cohen, C. L. Fry, and T. Rodriguez, "Picture my voice: Audio to visual speech synthesis using artificial neural networks," in *Proc. AVSP*, D. W. Massaro, Ed., Santa Cruz, CA, 1999, pp. 133–138.

[10] P. Hong, Z. Wen, and T. S. Huang, "Real-time speech-driven face animation with expressions using neural networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 4, pp. 916–927, Jul. 2002.

[11] T. Chen, "Audiovisual speech processing: Lip reading and lip synchronization," *IEEE Signal Process. Mag.*, vol. 18, no. 1, pp. 9–21, Jan. 2001.

[12] K. Choi, Y. Luo, and J.-N. Hwang, "Hidden Markov model inversion for audio-to-visual conversion in an MPEG-4 facial animation system," *J. VLSI Signal Process.*, vol. 29, no. 1–2, pp. 51–61, 2001.

[13] M. Brand, "Voice puppetry," in *Proc. SIGGRAPH'99*, Los Angeles, CA, 1999, pp. 21–28.

[14] E. Yamamoto, S. Nakamura, and K. Shikano, "Lip movement synthesis from speech based on hidden Markov models," *Speech Commun.*, vol. 26, no. 1–2, pp. 105–115, 1998.

[15] R. R. Rao, T. Chen, and R. M. Mersereau, "Audio-to-visual conversion for multimedia communication," *IEEE Trans. Ind. Electron.*, vol. 45, no. 1, pp. 15–22, Feb. 1998.

[16] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[17] M. Brand, "Shadow puppetry," in *Proc. ICCV'99*, Corfu, Greece, Sep. 1999, pp. 1237–1244.

[18] B. N. Datta, *Numerical Linear Algebra and Applications*. Pacific Grove, CA: Brooks/Cole, 1995.

[19] S. Fu, "Audio/Visual Mapping Based on Hidden Markov Models," Master's thesis, Dept. Comput. Sci. and Eng., Wright State Univ., , Dayton, OH, 2002.

[20] K. Choi and J.-N Hwang, "Baum-Welch HMM inversion for audio-to-visual conversion," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, 1999, pp. 175–180.

[21] S. Moon and J.-N. Hwang, "Robust speech recognition based on joint model and feature space optimization of hidden Markov models," *IEEE Tran. Neural Netw.*, vol. 8, no. 2, pp. 194–204, Mar. 1997.

[22] L. E. Baum and G. R. Sell, "Growth functions for transformations on manifolds," *Pacific J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[23] Y. Li and H.-Y. Shum, "Learning dynamic audio/visual mapping with input-output hidden Markov models," in *Proc. 5th Asian Conf. on Computer Vision*, Melbourne, Australia, Jan. 2002.

[24] M. Brand, "Structure learning in conditional probability models via an entropic prior and parameter extinction," *Neural Comput.*, vol. 11, no. 5, pp. 1155–1182, 1999.

[25] F. I. Parke and K. Waters, *Computer Facial Animation*. Wesley, MA: A.K Peters, 1996.

[26] P. K. Kakumanu, "Audio-Visual Processing for Speech Driven Facial Animation," Master, Dept. Comput. Sci. and Eng., Wright State Univ., , Dayton, OH, 2002.

[27] D. O'Shaughnessy, *Speech Communication: Human and Machine*, 2nd ed. New York: IEEE, 2000.

[28] G. Aversano, A. Esposito, A. Esposito, and M. Marinaro, "A new text-independent method for phoneme segmentation," in *Proc. 44th IEEE Midwest Symp. Circuits and Systems*, vol. 2, 2001, pp. 516–519.

[29] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Boston, MA: Academic, 1990.

[30] R. Koenen, "Overview of the MPEG-4 Standard," ISO/IEC JTC1/SC29/WG11, Seoul, South Korea, 1999.

[31] R. Gutierrez-Osuna, P. K. Kakumanu, A. Esposito, O. N. Garcia, A. Bojorquez, J. L. Castillo, and I. J. Rudomin, "Speech-driven facial animation with realistic dynamics," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 33–42, Feb. 2005.

[32] J. S. Garofolo, *Getting Started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database*. Gaithersburg, MD: NIST, 1988.

Shengli Fu (S'03) received the B.S. and M.S. degrees in telecommunications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively, and the M.S. degree in computer engineering from Wright State University, Dayton, OH, in 2002. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Delaware, Newark.

His research interests include acoustic and visual signal processing, information and coding theory.

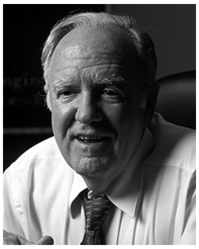
Ricardo Gutierrez-Osuna (M'00) received the B.S. degree in electronics engineering from the Polytechnic University of Madrid, Madrid, Spain, in 1992, and the M.S. and Ph.D. degrees in computer engineering from North Carolina State University, Raleigh, in 1995 and 1998, respectively.

From 1998 to 2002, he served on the faculty at Wright State University, Dayton, OH. He is currently an Assistant Professor of Computer Engineering at Texas A&M University, College Station. His research interests include pattern recognition, machine olfaction, biological cybernetics, speech-driven facial animation, computer vision, and mobile robotics.

Anna Esposito is an Associate Professor in the Department of Psychology, Second University of Naples, Naples, Italy. She is also affiliated with the International Institute for Advanced Scientific Studies (Italy) and the Speech Communication Group, MIT, Cambridge, MA. From 1999 to 2002, she was a Research Associate at Wright State University, Dayton, OH. Her research interests include speech segmentation, acoustic features of visual speech, cross-modal analysis of speech, gestures, and gaze, and neural networks.

Praveen K. Kakumanu (S'03) received the M.S. degree in computer science in 2002 from Wright State University (WSU), Dayton, OH, where he is currently pursuing the Ph.D. degree in computer science.

He is an Instructor at WSU. His research interests are in the areas of human-computer interaction, speech animation, and pattern recognition.



Oscar N. Garcia (M'58–SM'71–F'84) received the B.S. and M.S. degrees from North Carolina State University, Raleigh, and the Ph.D. degree from the University of Maryland, College Park.

He is currently Founding Dean of Engineering at the University of North Texas, Denton. Previously, he was NCR Distinguished Professor and Chair in the Computer Science and Engineering Department, Wright State University, Dayton, OH. He served the National Science Foundation (NSF)as Program Officer in the CISE and EHH Directorates for several years. He was a Full Professor of Electrical Engineering and Computer Science at the George Washington University, Washington, DC, and has been Charter Chair of the Department of Computer Science and Engineering at the University of South Florida, St. Petersburg.

Dr. Garcia is a Past President of the IEEE Computer Society and has served on the IEEE Board of Directors. He is currently Secretary of the Society, Chairman of its Awards Committee and a Member of its Board of Governors. He received the Merwin Award of the Computer Society and the Emberson Award of the IEEE. He is a Fellow of the American Association for the Advancement of Science.