

# Accurate Visible Speech Synthesis Based on Concatenating Variable Length Motion Capture Data

Jiyong Ma, *Member, IEEE*, Ron Cole, *Member, IEEE*, Bryan Pellom, *Member, IEEE*, Wayne Ward, and Barbara Wise

**Abstract**—We present a novel approach to synthesizing accurate visible speech based on searching and concatenating optimal variable-length units in a large corpus of motion capture data. Based on a set of visual prototypes selected on a source face and a corresponding set designated for a target face, we propose a machine learning technique to automatically map the facial motions observed on the source face to the target face. In order to model the long distance coarticulation effects in visible speech, a large-scale corpus that covers the most common syllables in English was collected, annotated and analyzed. For any input text, a search algorithm to locate the optimal sequences of concatenated units for synthesis is described. A new algorithm to adapt lip motions from a generic 3D face model to a specific 3D face model is also proposed. A complete, end-to-end visible speech animation system is implemented based on the approach. This system is currently used in more than 60 kindergarten through third grade classrooms to teach students to read using a lifelike conversational animated agent. To evaluate the quality of the visible speech produced by the animation system, both subjective evaluation and objective evaluation are conducted. The evaluation results show that the proposed approach is accurate and powerful for visible speech synthesis.

**Index Terms**—Face animation, character animation, visual speech, visible speech, coarticulation effect, virtual human.

## 1 INTRODUCTION

SYNTHESIS of realistic, accurate visible speech is an active research area that has many applications within virtual humans, including improved intelligibility of speech in noisy environments and learning, training and even certain types of therapy [1], [2]. To achieve accurate visible speech synthesis, one important task is to model coarticulation effects that occur when speech segments are influenced by prior or subsequent segments; such effects can occur across several phonemes, such as lip rounding that may occur during the /s/ in the word “strewn” [3]. A direct way to model coarticulation effects is to collect motion capture data from a person’s lips and lower face during speech production and map the facial movements points onto a 3D face model. The captured data could then be used to synthesize visible speech for different face models.

Based on this idea, we developed a visible synthesis system in our previous research [4]. In this system, all transition movements from one viseme to another were captured to simulate the coarticulation effects from adjacent phonemes. During the synthesis of visible speech of 3D models from text, these viseme units were located in a database, concatenated, and processed to conform to the durations of the speech segments in the speech, and then used to move the lips and lower face regions of the model.

- The authors are with the Center for Spoken Language Research, University of Colorado at Boulder, Campus Box 594, Boulder, CO 80309-0594. E-mail: {jiyong, cole}@cslr.colorado.edu.

Manuscript received 3 Feb. 2005; revised 4 Apr. 2005; accepted 17 May 2005; published online 10 Jan. 2006.

For information on obtaining reprints of this article, please send e-mail to: [tcvg@computer.org](mailto:tcvg@computer.org), and reference IEEECS Log Number TVCG-0012-0205.

While this approach produced natural and accurate visible speech in most contexts, it is not capable of modeling long distance coarticulation effects (that can extend over several prior or subsequent phonemes), as the unit of concatenation was limited to the diviseme.

In order to model long-distance coarticulation effects, we extend the diviseme approach to concatenation and processing of the most frequently occurring words and syllables in English. The idea behind this approach is to locate and concatenate the longest sequence of segments (which could be a word), as is done today in the most successful domain-independent text-to-speech systems. In addition, an example-based machine learning approach is applied to estimate the motion mapping from the captured data to a target 3D face model.

In Section 2, we review previous work in face animation. Section 3 presents our approach to visible speech synthesis. Section 4 presents the results. Section 5 summarizes the work.

## 2 PREVIOUS WORK

Since Parke’s pioneering work [5] on face animation, several approaches have been proposed to create face animation, including parameter-based approaches [5], physics-based approaches [6], image-based approaches [7], and multi-target morphing approaches [8], [9], [10]. In the parameter-based or physics-based approach, a face model is parameterized with geometrical or physical parameters. Various parameters have been investigated such as the abstract muscle parameters proposed by Magnenat-Thalmann et al. [11], the parameters proposed in MPEG-4 [12], or the

blending of coefficients in the multitarget morphing approach.

Visible speech synthesis is an important component of face animation. There are three ways to synthesize visible speech: One is the rule-based approach in which the visible speech is generated by applying a set of rules and smoothing techniques [10], [12], [13], [14], [15], [16]. The second is the performance-driven approach [17] in which the visible speech is driven by a real person's facial movements measured by a motion capture system. The third is the speech-driven approach in which visible speech is produced by mapping speech feature vectors. For example, Jiang et al. [18] showed that the relationship between speech acoustics and facial optical movements was not uniform across talkers and vowel context. These results showed that it is a challenging task to synthesize accurate visible speech directly driven by auditory speech.

Motion capture data can be applied to different approaches to estimate the parameters for face animation. For instance, Kshirsagar et al. [19] used motion capture data to estimate facial animation parameters defined in MPEG4 standard.

The motion capture data can also be employed to estimate the parameters in the dominance functions proposed by Cohen and Massaro [20] or the parameters in the kernel functions proposed in our previous research [10]. In addition, motion capture data can also be used to estimate the blending coefficients in the multitarget morphing approach, the shape blending approach [8], [9], [10], [21], [22], or the linear combination approach [23].

In early work on performance-driven face animation [17], the motion mapping was implemented through a simple geometrical deformation algorithm. An extension of this approach can be found in [24]. Another extension to the geometrical deformation approach is the use of the RBF interpolation algorithm [25], [26], [27], [28]. Image-based approaches also have a connection with motion capture technology. In the image-based approach, the captured video data can be viewed as a special type of motion capture data in which the source face and the target face are the same. The motion-mapping task in this approach is to find the point correspondence in the two images captured at different times.

For example, Ezzat et al. [29] proposed an approach to visible speech synthesis based on a training video corpus. Recently Geiger et al. [30] reported the results of a perceptual evaluation test about the animation system mentioned above. The visible speech intelligibility test suggests that the image-based animation system still needs to be improved for the purposes of rehabilitation and language learning tasks. Some open issues in visible speech synthesis are reviewed in [31].

### 3 VISIBLE SPEECH SYNTHESIS

#### 3.1 System Overview

The system architecture of the visible speech synthesis system is shown in Fig. 1. The corpus consists of motion trajectories of 3D facial markers captured by a motion capture system. The recorded corpus of motion capture

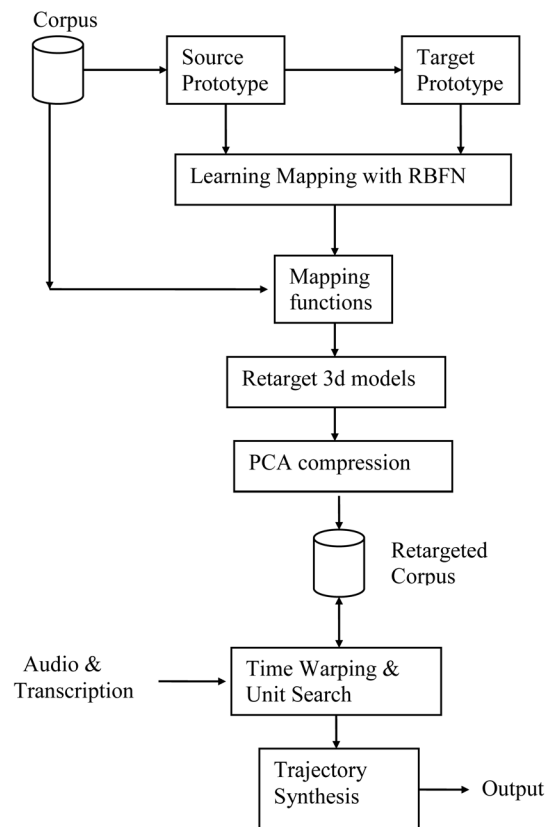


Fig. 1. The system architecture.

data is first processed. Then, a set of visual prototypes in the source face is selected, and its corresponding visual prototypes in the target 3D face are designed manually to enable each visual prototype designed for the target face to resemble to that in the source face.

The mapping functions from the source face to the target face are learned by Radial Basis Function Networks [32]. Then, the principal component analysis (PCA) is used to compress the 3D meshes generated by the motion capture data. A retargeted corpus is formed as a set of compact representations of the retargeted models with PCA. For each sequence in the motion capture database, a time-aligned phonetic transcription of the corresponding speech utterance was generated automatically using a speech recognizer and the text corresponding to the utterance.

The speech wave file relevant to the input text is generated either by a speech synthesizer or by recording audio with a microphone. A search algorithm is applied to find the best-concatenated units in the retargeted corpus. The time warping algorithm is applied to the selected units so that their durations conform to the duration of the generated phonetic string. A trajectory-smoothing algorithm is applied to get smooth-concatenated visible speech. The output is the optimal visible speech synchronized with auditory speech signal.

#### 3.2 Corpus

We collected motion capture data consisting of about 1,400 utterances at a motion capture studio in California. The word sequences, uttered by a professional speaker (a reading researcher with training in linguistics) contained the



Fig. 2. Original 3D facial markers.

most common single syllable words occurring in spoken English, as well as multisyllabic words containing the most common initial, medial, and final syllables of English. The frequency of occurrence of words and syllables was computed from a large corpus of annotated speech called the DARPA broadcast news HUB4 collected by the Linguistic Data Consortium LCD (<http://www ldc.upenn.edu/About/>). The corpus contains hundreds of stories and millions of words. The frequency of each word in the corpus of 64k different words is estimated from the text transcriptions containing millions of words. To estimate the frequency of each syllable in English, a syllabification system was designed based on the Festival speech synthesis system [33]. The English lexicon that contains 64k words is input to the system, which automatically segments each word into syllables. Then, the frequency of each syllable is estimated.

About 800 words, which cover the syllables with high occurring frequency, were selected. In addition, we selected the 100 most common words in English. Finally, to assure complete coverage of all divisemes, we recorded 400 nonsense syllables containing all of the divisemes of English. A diviseme is defined as a sequence of two visemes, and a viseme is defined as a visual production of one or more phonemes. To summarize, the corpus consists of the most frequently occurring syllables in multisyllabic words, frequent words, and a set of nonsense syllables representing individual divisemes. The proposed visible synthesis system is designed to locate and concatenate utterances by selecting the fewest representations in the corpus—i.e., representations that contain the longest motion capture sequences that will present all of the visemes in the target utterance.

The stimuli were recorded from 50 points on the speaker’s face and lips at 60 frames per second, as shown in Fig. 2, the facial marker’s positions defined in the MPEG4 standard [12] are used as the reference to determine these markers’ positions. Some markers are used to estimate the head pose. The motion capture session lasted approximately 8 hours. About 60k frames of motion capture data were collected. Only facial markers in the lower face are used in this research of visible speech production.

### 3.3 Preprocessing

The data were preprocessed to assign phonetic segment labels and boundaries to the motion capture sequences and to estimate the head pose of each frame of motion capture data in order to remove the influence of head movements

on visible speech synthesis. Time aligned phonetic segmentation was performed by the SONIC speech recognizer [34], which provided the phoneme sequences derived from text representations of each utterance. The pose estimation algorithm described in [35] was used to estimate head pose.

### 3.4 Prototype Selection

The corpus contained approximately 60,000 frames of motion capture data. Among them, we want to manually select some prototypes that represent the typical lip shape configurations. These prototypes will serve as the learning examples to design their corresponding prototypes in the target face model. The prototypes in both source face and target face will also be used to learn the mapping functions. Theoretically speaking, the more the prototypes are used, the higher the accuracy of the mapping functions is. However, increasing the number of prototypes will also increase the amount of work to manually design prototypes for the target face. Therefore, an appropriate number of prototypes should be determined. The choice of the number of prototypes should balance the accuracy of mapping functions and the amount of work required to design the target prototypes. Twenty-one visual prototypes are used to construct the mapping from the source face to the target face as shown in Fig. 3. The 3D tongue model is not included in the visual prototypes; we have a parameter-based tongue model that is discussed in our previous research [10]. The lower tooth 3D model in our system is also a parameter-based model that is controlled by the jaw rotation angle, i.e., the feature point 2.1 defined in MPEG4 [12].

### 3.5 Retargeting Motion

Estimating the mapping functions from the motion capture data to a target face model is an essential task. The learning examples, i.e., the prototypes selected in the source face and those designed for the target face model, allow us to construct the mapping functions. Many machine learning approaches could be used for this task. In this implementation, Radial Basis Function Networks (RBFN) [32] are chosen to construct the mapping functions.

Denote the prototypes selected in the source face as  $S_i, i = 0, 1, 2, \dots, m - 1, S_i \in R^{3p}$ , where  $p$  is the number of the measured 3D facial points on the subject’s lower face, and the prototypes designed for the target face model as  $T_i, T_i \in R^{3N}, i = 0, 1, 2, \dots, m - 1$ .  $N$  is the total number of vertices in the target face model and  $m$  is the number of visual prototypes.



Fig. 3. (a) Visual prototypes in the source face. (b) Visual prototypes designed for the target face.

RBFN can be expressed as the following [32]:

$$f(x) = \sum_{j=0}^{m-1} w_j h_j(x), \quad (1)$$

where  $h_j(x) = \exp(-\|x - S_j\|^2/r^2)$  are basis functions and  $\{w_j\}$  are the unknown coefficients to be estimated.  $f(x)$  is the mapping function. The parameter  $r^2$  is chosen as the variance of all captured samples  $\{x\}$ , i.e., the mean value of  $\|x - \bar{x}\|^2$ , where  $\bar{x}$  is the mean value of  $\{x\}$ .

The learning examples are denoted as  $\{S_j, u_j\}_{j=0}^{m-1}$ , where the vector  $S_j$  is a prototype defined for the source face and  $u_j$  is a coordinate component of the prototype  $T_j$  defined for the target face, i.e.,  $u_j$  is a list of  $3 \times N$  coordinates of the prototype  $T_j$  defined for the  $j$ th morph target of the target face.

Denote  $y = (u_0, u_1, \dots, u_{m-1})^t$ ;  $w = (w_0, w_1, \dots, w_{m-1})^t$ ;  $H = (h_j(S_i))$  as the design matrix. The fitting error is expressed as

$$e = y - Hw.$$

In order to find a robust solution of the coefficient vector  $w$ , the following squared-error is defined:

$$E = \|y - Hw\|^2 + \lambda \|w\|^2.$$

The second term in the right of above the equation is a penalty term.  $\lambda$  is the regularization parameter controlling the amount of penalty. In order to find the best regularization parameter  $\lambda$ , GCV (generalized cross-validation) is used as an objective function [32]. The mapping function defined in (1) is applied to different coordinates of all

vertices in the target face model. After the coefficients are estimated, (1) is used as the mapping functions for all vertices.

It is worth noting that the purpose of using the RBF approach in [26], [28] is different from the purpose of using RBF in this article. RBF is applied in [26], [28] for 3D model adaptation or deformation, whereas the RBF approach in this article is to map the source visual prototypes to their target counterparts. From a mathematical point of view, the main difference of the RBF approach proposed in this paper from RBF in [26], [28] is: The variable  $x$  used in RBF in this paper is a vector belonging to  $R^{3p}$ , whereas the variable  $x$  used in RBF in [26], [28] is a 2D or 3D vector. The motion mapping approach discussed in this article is also different from the approach proposed in our previous research [4] in which no RBF was applied.

### 3.6 Data Compression

The RBF approach maps each frame of motion capture data to a high-dimensional vector in  $R^{3N}$ , where  $N$  is the total number of vertices in the target face model. This creates many retargeted data from the motion capture data because there are a total of about 60,000 frames of motion capture data. One key issue is how to compress and access these frames efficiently. We selected PCA as a compression technique because PCA allows each frame of the retargeted 3D model data to be represented by a set of low-dimensional parameters.

In the PCA technique, a basis is computed with the retargeted high dimensional vectors. Then, a high-dimensional vector is projected on the basis; the projection coordinates are used as a compact representation of the retargeted face model. However, the traditional algorithm to do PCA requires computing the auto-covariance matrix. An alternative approach without explicit computation of the sample covariance is the expectation-maximization (EM) PCA learning algorithm (EM-PCA) [36], which is an iterative procedure for finding the subspace spanned by the leading eigenvectors.

When the dimension of vectors is very large, an efficient way to implement the EM-PCA algorithm is presented in the following. Let  $y_t$  be one frame of the retargeted 3D model data. The general linear model assumes that the set of observed  $p$ -dimensional vectors  $\{y_t\}$  is generated by a set of corresponding  $k$ -dimensional latent variables  $\{x_t\}$  by the equation:  $y_t = Cx_t + \varepsilon_t$ , where  $C$  is a  $p \times k$  matrix and  $\varepsilon_t$  is a  $p$ -dimensional noise vector. Denoting all observed data by a  $p \times n$  matrix  $Y = [y_1, y_2, \dots, y_n]$ , and all unknown latent variables by a  $k \times n$  matrix  $X = [x_1, x_2, \dots, x_n]$ . The EM-PCA algorithm as the following: 1) estimate  $X$  by the E-step:  $X = (C^t C)^{-1} C^t Y$  and 2) estimate  $C$  by the M-step:  $C^{new} = Y X^t (X X^t)^{-1}$ . The above two procedures work iteratively to refine the estimation of  $X$  and  $C$ . The columns of  $C$  will span the subspace of the first  $k$  principle components. In addition, this algorithm does not need to load all high-dimensional data into memory at the same time. For instance, each high-dimensional vector  $y_t$  and some elements in matrix  $X$  can be loaded into the memory at the same time. After the matrix  $C$  is estimated, each vector  $y_t$  is transformed into a low dimensional vector  $x_t$  by the equation  $x_t = (C^t C)^{-1} C^t y_t$ . In our implementation, the number of leading eigenvectors is set to 17.

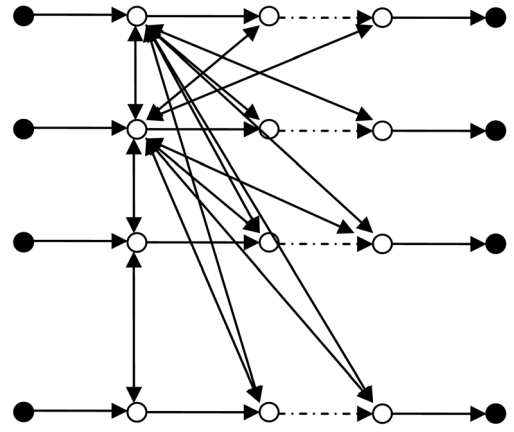


Fig. 4. The phonetic transition network, where each horizontal subgraph represents one phonetic string corresponding to one sequence in the motion capture data.

### 3.7 Concatenation

Concatenative approaches have been widely used for auditory speech synthesis. Previous research in speech synthesis has shown that longer units are desirable because they result in fewer concatenation discontinuities [37], [38]. Generally speaking, given the target phonetic specification from the front-end of a text-to-speech system, a unit selection algorithm needs to search through the speech corpus to find the optimal concatenated units that result in high quality synthetic speech. The selected units should: 1) concatenate well at boundaries, 2) be as long as possible, and 3) have similar context in the target specification. Hunt and Black [37] proposed a phoneme-based unit-searching algorithm, where the Viterbi search algorithm is applied to select the best units. A discussion of variable-length units for speech synthesis can be found in [38]. Cosatto and Graf [39] borrowed the unit-search algorithm [37] to visible speech synthesis. However, there is no detailed description about the Viterbi algorithm in [37], [38], [39]. In this article, we present a detailed unit selection algorithm for visible speech synthesis in which the concatenation cost is also different from previous research work in [39].

We now present our graph search technique for finding optimal units to concatenate to produce natural visible speech. All phonetic strings in the motion capture data are represented by a graph. There are about 1,400 words in the motion capture data. A state/node in the graph is represented by a unique index, and each state is associated with a phonetic symbol. The phonetic symbol string of each word in the corpus represents a directed subgraph. The states associated with phonetic strings of all words in the corpus can constitute a directed graph. In the graph, each horizontal subgraph represents the word corresponding to one sequence in the motion capture data, while for two states/nodes in different horizontal subgraphs, the two states/nodes may have a bidirectional connection.

In Fig. 4, the solid dots represent the start or the end state in a word. The circles represent the state between the start and end states. The connections between two nodes/states represent the transition from one phoneme to another.

An input text is transcribed into a target phonetic specification using the front-end of text-to-speech system.

The target phonetic specification represents the phonetic strings corresponding to the input text. Given the observed phonetic sequence, the task is to find the best-concatenated units in the graph that match the given target specification. Because the optimal states corresponding to the optimal units are unknown, the states are called hidden states.

The *Viterbi* algorithm [40] is a dynamic programming algorithm that finds the most likely sequence of hidden states, known as the *Viterbi* path, which generates the sequence of observed events, especially in the context of hidden *Markov* models.

Suppose that there are  $N$  states in phonetic transition network shown in Fig. 4. Denote the observed phonetic sequence as  $O = \{o_1, o_2, \dots, o_T\}$ , where  $T$  is the total number of phonemes. Denote the concatenation cost from state  $i$  to state  $j$  as  $c_{ij}$ . The state observation cost is denoted as  $d_i(o_t)$ , i.e., the cost that the state  $i$  generates the observed event  $o_t$ .

In order to find the best hidden state sequence,  $Q = \{q_1, q_2, \dots, q_T\}$ , the *Viterbi* algorithm is described as the following [40]:

1. Initialization:

$$\varphi_1(i) = d_i(o_1); 1 \leq i \leq N.$$

2. Recursion:

$$\varphi_t(j) = \min_i \{\varphi_{t-1}(i) + c_{ij} + d_j(o_t)\}; 1 \leq j \leq N;$$

$$2 \leq t \leq T$$

$$\psi_t(j) = \arg \min_i \{\varphi_{t-1}(i) + c_{ij} + d_j(o_t)\};$$

$$1 \leq j \leq N; 2 \leq t \leq T.$$

3. Termination:

$$P^* = \min_i \{\varphi_T(i)\}; q_T^* = \arg \min_i \{\varphi_T(i)\}.$$

4. Path backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \dots, 1$$

$$\varphi_t(j) = \min_i \{\varphi_{t-1}(i) + c_{ij} + d_j(o_t)\}.$$

From the optimal state sequence  $Q^* = \{q_1^*, q_2^*, \dots, q_T^*\}$ , we can find the optimal concatenated units.

It can be seen that the *Viterbi* algorithm described above is an extension of the algorithm to find the least cost path, because there are additional constraints used in the *Viterbi* algorithm, i.e., the state observation costs  $d_i(o_t)$ .

In the following discussion, we say that state  $i$  and state  $j$  belong to the same viseme category, if and only if the phoneme associated with state  $i$  and the phoneme associated with state  $j$  belong to the same viseme category.

For two connected nodes/states in the graph such as  $i, j$ , if they are from the same horizontal subgraph, i.e., they are adjacent in the motion capture data, the connected cost is defined as zero, i.e.,  $c_{ij} = 0$ , because the motion from one unit to another is already recorded in motion capture data,

we can directly use the unit in motion capture data without need of concatenation.

For two connected states  $i$  and  $j$  in the graph, if they are not from the same horizontal subgraph, i.e., they are not in the same captured sequence, the admissible connection condition for the state  $i$  and  $j$  is that the parent state of state  $j$  should have the same viseme with the state  $i$ , i.e., they belong to the same viseme category, because the actual concatenation boundary is between the state  $i$  and the parent state of state  $j$ . Therefore, the concatenation cost of state  $i$  and state  $j$  is defined as  $c_{ij} = 1 + \beta (C(i, \text{parent}(j)) + G(i, \text{parent}(j)))$ , where  $\text{parent}(j)$  is the parent state of state  $j$ .  $\text{parent}(j)$  and  $j$  belong to the same horizontal subgraph, and state  $\text{parent}(j)$  precedes state  $j$ . For instance, if the state sequence corresponding to a horizontal subgraph is /abcd/, if  $c = j$ , then  $b = \text{parent}(j)$ .  $\beta$  is a nonnegative constant. When  $\beta = 0$ , the optimal solution is the minimum number of concatenated segments. The concatenated cost  $C(i, \text{parent}(j))$  is defined as the degree of smoothing of the trajectory in the transition period from one state to another. It is defined as:

$$C(i, \text{parent}(j)) = \int_{t_0}^{t_1} \|S^{(2)}(t)\|^2 dt / (t_1 - t_0),$$

where  $t_0$  and  $t_1$  represent the start time of state  $i$  and the end time of state  $\text{parent}(j)$ , respectively, and  $S(t)$  is the trajectory blended by motion vectors of state  $i$  and state  $\text{parent}(j)$ . Note that a motion vector is the low-dimensional vector  $x_t$  defined in the Section 3.6. The detailed blending algorithm can be found in the motion vector blending section in our previous research [4].

If state  $\text{parent}(j)$  and state  $i$  belong to the same viseme category, then  $G(i, \text{parent}(j)) = 0$ , and this means that state  $i$  can be connected with the state  $j$ , because its parent state in the captured sequence has the same viseme with state  $i$ . If state  $\text{parent}(j)$  and state  $i$  do not belong to the same viseme category, then  $G(i, \text{parent}(j)) = +\infty$ . The viseme category is defined in Fig. 3 and in [4].

If the phonemes associated with state  $j$  and  $o_t$  belong to the same viseme category, the state observation cost is defined as zero, i.e.,  $d_j(o_t) = 0$ , otherwise,  $d_j(o_t) = +\infty$ .

The computation complexity of the *Viterbi* algorithm in full search mode is  $N \times N \times T$  [40]. However, in practical applications, because a lot of states at each frame are inactive, the search space is quite small. For instance, if the state observation cost is a positive infinite quantity, i.e.,  $d_j(o_t) = +\infty$ , or the concatenation cost is a positive infinite quantity, i.e.,  $c_{ij} = +\infty$ , the state  $j$  will be inactive and it will be pruned during the search process.

Actually, the number of active states at each frame  $t$  is very small. An active state  $j$  at frame  $t$  must satisfy the following two conditions: 1) The phoneme associated with state  $j$  has the same viseme with  $o_t$  and 2) its parent state  $\text{parent}(j)$  has the same viseme with the state  $i$ ; this condition ensures that the state  $i$  can be concatenated with  $j$ . The search algorithm is very fast because the search space is relatively small at each frame. A threshold such as the total number of active states can also be employed to control how many states will participate in the search at each frame to speed up the decoding speed.



Fig. 5. (a) Marni's model. (b) Pavarotti's model.

In comparison, the proposed unit selection algorithm is different from the algorithm proposed in [39], [41]; the differences between these two algorithms lie in: 1) different concatenation costs and 2) different search algorithms. In the concatenation cost, we did not include the speech signal; the reason is that spectral information extracted from the speech signal may not provide sufficient information to determine a realistic synthetic visible speech sequence. For example, the acoustic features of the speech segments /s/ and /p/ in an utterance of the word “spoon” are quite different from those for the phoneme /u/, whereas the lip shapes of /s/ and /p/ in this utterance are very similar to the phoneme /u/. The concatenation cost defined in this article includes the smoothing term of two concatenated units. Our search algorithm is based on the framework of the *Viterbi* algorithm used in speech recognition that is a special case of dynamic programming in mathematics. A more comprehensive discussion of the search algorithm can be found in [42].

### 3.8 Trajectory Smoothing

Abrupt changes in the juncture of two units occur because the lip shapes may be different for the same phoneme spoken in different contexts even though a low pass filter has already been applied to make the transition motion smooth. The aim of trajectory smoothing is to make the lip motion more natural and smooth. A trajectory-smoothing algorithm [4], [43] proposed in our previous research is applied to make the concatenated trajectory smooth.

### 3.9 Model Adaptation

Model adaptation is necessary to apply the visible speech synthesis approach to new 3D models. The automatic adaptation process saves the time required to design morph targets for the specific 3D face model and will map the visible speech produced by the generic model to that of the specific 3D face model. We achieve this by deforming the generic 3D generic model to match the targets of a specific 3D target model. For instance, as shown in Fig. 5, Marni's 3D model can be viewed as a generic model with a set of designed morph targets, such as facial expression morph targets and viseme targets, whereas Pavarotti's 3D model is a specific model.

The following will address the problem of adapting the motions and morph targets of Marni's model to those of Pavarotti's model.

Noh and Neumann [26] proposed an approach to 3D model adaptation based on the coordinate transformation of

the local coordinate systems defined in the source face model and in the target face model. The approach proposed in this article is different from the above approach [26]. Our approach is more simple and intuitive. The approach computes the transformation directly from the corresponding 3D meshes in the generic face model and the retargeted model. Because all vertices' positions of the generic model and the specific model are known, an affine transformation is estimated from the two sets of data. The affine transformation includes transformations such as scaling, rotation, and translation.

The affine transformation mapping one triangular polygon in the generic model to its corresponding polygon in the specific model can be estimated by the following interpolation algorithm:

$$y_p = Ax_p; \text{ for } p = i, j, k. \quad (2)$$

Fig. 6 shows the geometrical relationship of these vectors, where  $y_p = \tilde{v}_p - \tilde{v}_C$ ,  $x_p = v_p - v_C$ , and  $\tilde{v}_C, v_C$  are the two reference 3D points/vectors selected for the specific model and generic model, respectively (we do not distinguish the difference between points and vectors, because they have a unique correspondence relationship once the coordinate system is selected). The vectors are selected at the centers of the two models, respectively.  $i, j, k$  are vertex indices of the three vertices on the triangle polygon.  $\tilde{v}_p, v_p$  are position vectors of the vertices in the specific model and the generic model, respectively.  $A$  is the affine transformation matrix to be estimated. Three equations in (2) can determine a unique affine transformation if the three vectors  $x_p = v_p - v_C$  ( $p = i, j, k$ ) are not located in a plane. This condition can be met for most triangular

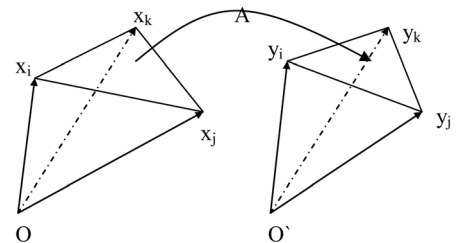


Fig. 6. One triangle polygon in the generic 3D model is mapped onto its corresponding triangle polygon in the retargeted model with affine transformation  $A$  that satisfies the equations:  $y_p = Ax_p$ , where  $y_p = \tilde{v}_p - \tilde{v}_C$ ,  $x_p = v_p - v_C$ , and  $p = i, j, k$ .

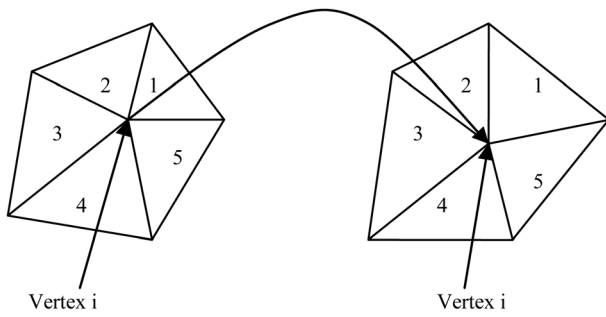


Fig. 7. The vertex  $i$  in the generic 3D model is mapped onto the vertex  $i$  in the specific 3D model. There are five triangle polygons neighboring the vertex  $i$ .

polygon meshes. For a triangular polygon, if this condition cannot be met, this triangular polygon is irregular. A triangle polygon mesh is irregular if the three vectors  $v_p$  ( $p = i, j, k$ ) consisting of the triangle polygon are located in one line, i.e., the area of the triangle mesh is zero. The irregular polygon mesh can also be regularized by moving one vector to a position so that these three vectors are not in one line.

The affine transformation mapping a vertex in the generic model to its corresponding vertex in the specific model is defined as the weighted average of affine transformations associated with the triangular polygons neighboring this vertex. The affine transformation can be formulated as the following:

$$\hat{A}_i = \sum_{p \in N_i} s_p A_p / \sum_{p \in N_i} s_p, \quad (3)$$

where  $N_i$  denotes the set of triangular polygons neighboring vertex  $i$ .  $s_p, A_p$  denote the area of the triangular polygon  $p$  and the affine transformation associated with the triangular polygon  $p$ .  $\hat{A}_i$  denotes the affine transformation of the vertex  $i$ . When a triangle polygon is irregular, its area  $s_p$  is zero, therefore, the irregular polygon does not play any role in determining the affine transformation defined in (3). A geometrical relationship of the polygons adjacent to the vertex  $i$  is shown in Fig. 7.

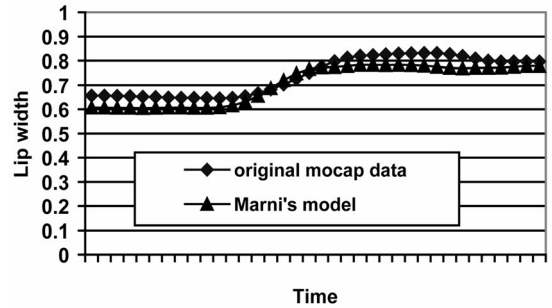
After each affine transformation associated with each vertex has been estimated by (3), the targets or the lip motions of the generic model can be adapted to the specific model with the equations  $\Delta \tilde{v}_i = \hat{A}_i \Delta v_i$ , where  $\Delta v_i$  and  $\Delta \tilde{v}_i$  are the displacements of the  $i$ th vertex in the generic model and in the specific model, respectively.

## 4 RESULTS

### 4.1 Objective Evaluation

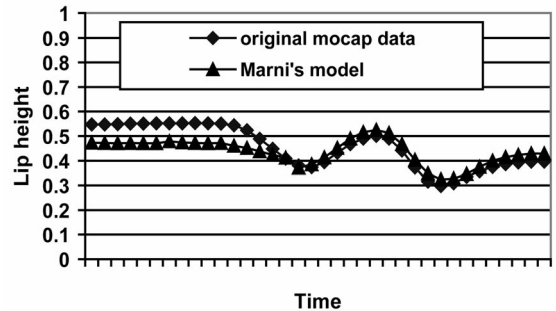
In the objective evaluation, the average errors between the normalized parameters in the source and the target model are used. These normalized parameters include: the lip height, lip width, and lip protrusion. The lip height is defined as the distance between two points on the centers of the upper lip and lower lip, respectively. The lip width is defined as the distance between two points at the lip corners. The lip protrusion is defined as the distance between the middle point in the upper lip and a reference point selected near jaw root.

In order to normalize the lip height, lip width, and lip protrusion, their maximum values are computed. The



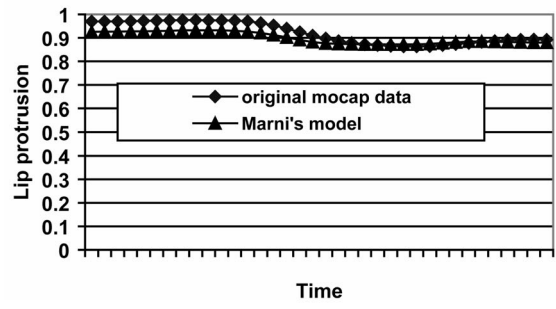
Time

(a)



Time

(b)



Time

(c)

Fig. 8. Comparison of lip parameter curves generated by original motion capture data and by Marni's model: (a) lip width, (b) lip height, and (c) lip protrusion.

maximum values of these three lip parameters in the source face model can be computed from all observed motion capture data, whereas the maximum values of these three lip parameters in the retargeted face model can be computed from the 3D model data derived by motion capture data.

The captured facial feature points shown in Fig. 2, which were mapped onto Marni's model shown in Fig. 5a, were used in the objective evaluation tests. The visible speech synthesis approach is applied to Marni's model. The average errors for lip height, lip width, and lip protrusion of Marni's model are 5.207 percent, 4.778 percent, and 2.21 percent. Fig. 8 illustrates the comparison of the lip parameter curves of the utterance "whomever" computed from original captured data and from Marni's model.

### 4.2 Subjective Evaluation

A subjective evaluation experiment of the visible speech was also conducted. Stimuli consisting of 16 sentences randomly selected from the TIMIT database are used in the

TABLE 1

The Average Accuracy and Naturalness of Sentence Level Visible Speech

	Real Speakers	System A	System B
Accuracy	4.30000	3.11667	2.93333
Naturalness	4.23333	3.06667	2.98333

experiment. The average values of the accuracy and naturalness of visible speech produced by real speakers and two animation systems are listed in the Table 1: System A represents the animation system based on the approach proposed in this paper and system B represents the animation system based on the di-viseme concatenation approach proposed in our previous research [4]. The accuracy and naturalness are measured by a Mean Opinion Score (MOS) that is a subjective measure in a five level scale, i.e., 1(bad), 2(poor), 3(fair), 4(good), and 5(excellent). It can be seen that the accuracy and naturalness of the visible speech produced by the approach proposed in this paper is better than those of the di-viseme concatenation approach. However, the visible speech produced by real speakers is still better than that produced by the synthesis system.

### 4.3 Emotion Targets

In addition to applying the adaptation method described above to visible speech, we applied the approach to map emotion targets from Marni's 3D model to Pavarotti's model. Fig. 9 shows the six universal facial expression targets designed for Marni's model are successfully mapped onto Pavarotti's model.

We also created several video clips of visible speech produced by the approach proposed in this paper which can be downloaded on the following Web sites: <http://cslr.colorado.edu/~jyong/visiblespeech2.htm> and <http://cslr.colorado.edu/~jyong/TVCG-0012-0205demo.zip>.

## 5 SUMMARY

We have presented a new approach to visible speech synthesis. Based on this framework, a visible speech synthesis system has been designed and implemented. We have demonstrated new techniques for synthesizing accurate visible speech based on searching and concatenating variable length motion capture data. For motion mapping, our new contributions are the use of nonlinear mapping functions and the use of regularization parameters to reduce the distortions in the lip region caused by the mapping functions.

For the huge amount of retargeted motion data, PCA is applied to get a compact representation of the data that result in a fast deformation algorithm driven by motion capture data. In order to find the optimal variable-length units for an input text, a search algorithm under the framework of the *Viterbi* decoding algorithm used in speech recognition is proposed. Our new contributions to this approach are: A directed graph is proposed to represent all possible connections among the lexical units in the motion capture data, a cost is defined for each connection, and a detailed search algorithm is described. The search algorithm is very fast and efficient.

In order to adapt the morph targets and visible speech generated by a generic 3D face model to a specific 3D face model, a novel adaptation algorithm is proposed.

Based on the approach proposed in this paper, a visible speech synthesis system has been implemented. The animation system has also been integrated into CSLU toolkit that can be downloaded on the Web site: <http://cslr.colorado.edu/toolkit/main.html>. The system is currently being used to generate Marni's visible speech in a year long reading program that is deployed in 60 kindergarten through third grade classrooms in Colorado schools [3], [44]. In this program, Marni converses with children to teach them to read and learn from text. Marni is also being used in three different speech therapy programs to improve

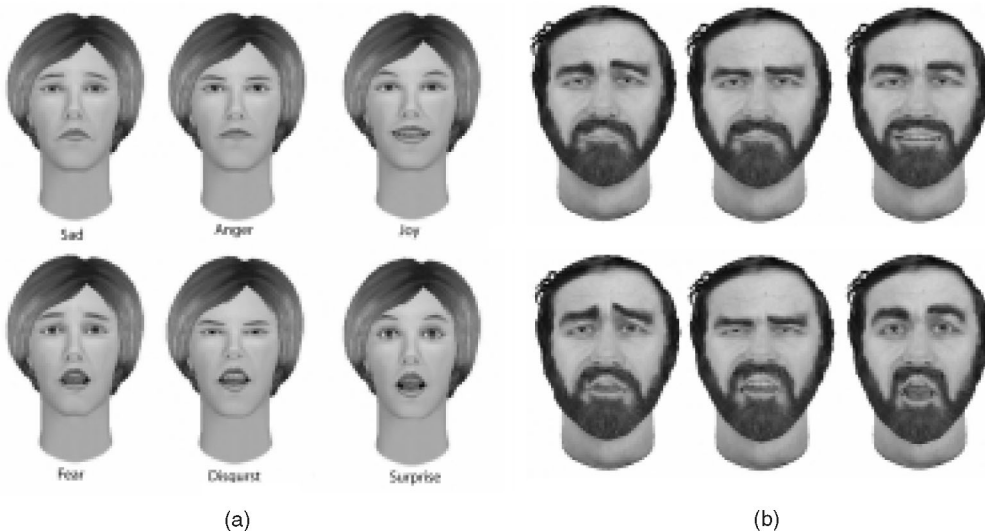


Fig. 9. (a) Six universal facial expression targets designed for Marni's model. (b) Six facial expression targets derived by mapping Marni's expression targets for Pavarotti's 3D model in Fig. 5.

the speech communication skills of individuals with Parkinsons disease and individuals with aphasia.

The experiment's results show that the system can produce more accurate visible speech. However, there is still room for improvement compared with the visible speech produced by human beings.

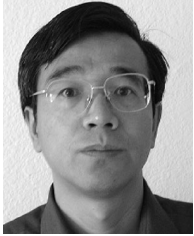
## ACKNOWLEDGMENTS

The authors would like to thank the editor and the reviewers for their comments to revise this article to make it better and better. This work was supported in part by US National Science Foundation (NSF) CARE grant EIA-9996075; NSF/ITR grant IIS-0086107; NSF/ITR Grant REC-0115419; NSF/IERI (Interagency Education Research Initiative) Grant EIA-0121201, and NSF/IERI Grant 1R01HD-44276.01. The findings and opinions expressed in this article do not necessarily represent those of the granting agencies.

## REFERENCES

- [1] R. Cole, D.W. Massaro, J. de Villiers, B. Rundle, K. Shobaki, J. Wouters, M. Cohen, J. Beskow, P. Stone, P. Connors, A. Tarachow, and D. Solcher, "New Tools for Interactive Speech and Language Training: Using Animated Conversational Agents in the Classrooms of Profoundly Deaf Children," *Proc. ESCA/SOCRATES*, 1999.
- [2] R. Cole, S. Van Vuuren, B. Pellom, K. Hacıoglu, J. Ma, J. Movellan, S. Schwartz, D. Wade-Stein, W. Ward, and J. Yan, "Perceptive Animated Interfaces: First Steps toward a New Paradigm for Human-Computer Interaction," *Proc. IEEE*, vol. 91, no. 9, pp. 1391-1405, 2003.
- [3] R.D. Kent and F.D. Minifie, "Coarticulation in Recent Speech Production Models," *J. Phonetics*, vol. 5, pp. 115-135, 1977.
- [4] J. Ma, R.A. Cole, B. Pellom, W. Ward, and B. Wise, "Accurate Automatic Visible Speech Synthesis of Arbitrary 3D Models Based on Concatenation of Diviseme Motion Capture Data," *J. Computer Animation and Virtual Worlds*, vol. 15, no. 5, pp. 485-500, 2004.
- [5] F. Parke, "Computer Generated Animation of Faces," *Proc. ACM Nat'l Conf.*, pp. 451-457, 1972.
- [6] D. Terzopoulos and K. Waters, "Physically-Based Facial Modeling, Analysis, and Animation," *J. Visualization and Computer Animation*, vol. 1, no. 4, pp. 73-80, 1990.
- [7] C. Bregler, M. Covell, and M. Slaney, "Video Rewrite: Driving Visual Speech with Audio," *Proc. ACM SIGGRAPH*, pp. 353-360, 1997.
- [8] C. Kouadio, P. Poulin, and P. Lachapelle, "Real Time Facial Animation Based upon a Bank of 3D Facial Expressions," *Proc. Computer Animation*, pp. 128-136, 1998.
- [9] J. Ma, J. Yan, and R. Cole, "CU Animate: Tools for Enabling Conversions with Animated Characters," *Proc. Int'l Conf. Spoken Language Processing*, pp. 197-200, 2002.
- [10] J. Ma and R. Cole, "Animating Visible Speech and Facial Expressions," *The Visual Computer*, vol. 20, nos. 2-3, pp. 86-105, 2004.
- [11] N. Magnenat-Thalmann, E. Primeau, and D. Thalmann, "Abstract Muscle Action Procedures for Human Face Animation," *The Visual Computer*, vol. 3, no. 5, pp. 290-297, 1988.
- [12] I.S. Pandzic and R. Forchheimer, *MPEG-4 Facial Animation: The Standard, Implementation, and Applications*. John Wiley and Sons, Inc., 2002.
- [13] C. Pelachaud, N. Badler, and M. Steedman, "Linguistic Issues in Facial Animation," *Proc. Computer Animation*, pp. 15-30, 1991.
- [14] P. Cosi and G. Perin, "Labial Coarticulation Modeling for Realistic Facial Animation," *Proc. Int'l Conf. Multimodal Interfaces '02*, pp. 505-510, 2002.
- [15] J. Beskow, "Rule-Based Visual Speech Synthesis," *Proc. Eurospeech*, pp. 299-302, 1995.
- [16] L. Reveret and C. Benoit, "A New 3D Lip Model for Analysis and Synthesis of Lip Motion in Speech Production," *Proc. Second ESCA Workshop Audio-Visual Speech Processing*, Dec. 1998.
- [17] L. Williams, "Performance-Driven Facial Animation," *Proc. ACM SIGGRAPH Computer Graphics Conf.*, vol. 24, no. 4, pp. 235-242, 1990.
- [18] J. Jiang, A. Alwan, P. Keating, E. Auer, and L. Bernstein, "On the Relationship between Facial Movements, Tongue Movements, and Speech Acoustics," *EURASIP J. Applied Signal Processing*, special issue on joint audio-visual speech processing, vol. 11, pp. 1174-1188, 2002.
- [19] S. Kshirsagar, T. Molet, N. Magnenat-Thalmann, "Principal Components of Expressive Speech Animation," *Proc. Computer Graphics Int'l Conf.*, pp. 38-44, 2002.
- [20] M. Cohen and D.W. Massaro, "Modeling Coarticulation in Synthetic Visual Speech," *Proc. Computer Animation*, pp. 139-156, 1993.
- [21] P. Joshi, W.C. Tien, M. Desbrun, and F. Pighin, "Learning Controls for Blend Shape Based Realistic Facial Animation," *Proc. ACM SIGGRAPH Symp. Computer Animation*, pp. 187-192, 2003.
- [22] E. Chuang and C. Bregler, "Performance Driven Facial Animation Using Blendshape Interpolation," Technical Report CS-TR-2002-02, Computer Science Dept., Stanford Univ., year?
- [23] T. Vetter and T. Poggio, "Linear Object Classes and Image Synthesis From a Single Example Image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 733-742, 1997.
- [24] E.C. Patterson, P.C. Litwinowicz, and N. Greene, "Facial Animation by Spatial Mapping," *Proc. Computer Animation*, 1991.
- [25] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making Faces," *Proc. SIGGRAPH*, pp. 55-66, 1998.
- [26] J.Y. Noh and U. Neumann, "Expression Cloning," *Proc. ACM SIGGRAPH*, pp. 277-288, 2001.
- [27] M. Sanchez, J. Edge, S. King, and S. Maddock, "Use and Re-Use of Facial Motion Capture Data," *Proc. Vision, Video, and Graphics Conf.*, pp. 1-8, 2003.
- [28] F. Pighin, R. Szeliski, and D. Salesin, "Modeling and Animating Realistic Faces from Images," *Int'l J. Computer Vision*, special issue on video computing, vol. 50, no. 2, pp. 143-169, 2002.
- [29] T. Ezzat, G. Geiger, and T. Poggio, "Trainable Video Realistic Speech Animation," *Proc. ACM SIGGRAPH*, pp. 388-398, 2002.
- [30] G. Geiger, T. Ezzat, and T. Poggio, "Perceptual Evaluation of Video-Realistic Speech," CBCL Paper #224/AI Memo #2003-003, Mass. Inst. of Technology, Cambridge, Mass., Feb. 2003.
- [31] R. Parent, S. King, and O. Fujimura, "Issues in Lip-Sync Animation: Can You Read My Lips," *Computer Animation*, pp. 3-10, June 2002.
- [32] S.W. Choi, D. Lee, J.H. Park, and I.B. Lee, "Nonlinear Regression Using RBFN with Linear Sub Models," *Chemometrics and Intelligent Laboratory Systems*, vol. 65, no. 2, pp. 191-208, 2003.
- [33] The Festival Speech Synthesis System, <http://www.cstr.ed.ac.uk/projects/festival/>, 2006.
- [34] N. Pellom and K. Hacıoglu, "Recent Improvements in the SONIC ASR System for Noisy Speech: The SPINE Task," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 4-7, 2003.
- [35] R.M. Haralick, H. Joo, C. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim, "Pose Estimation from Corresponding Point Data," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1426-1446, 1989.
- [36] S. Roweis, "EM Algorithms for PCA and SPCA," *Advances in Neural Information Processing Systems*, vol. 10, pp. 626-632, 1998.
- [37] A.J. Hunt and A.W. Black, "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 373-376, 1996.
- [38] M. Lee, D.P. Lopresti, and J.P. Olive, "A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions," *Proc. ISCA Research Workshop Speech Synthesis*, pp. 347-356, Aug.-Sept. 2001.
- [39] E. Cosatto and H.P. Graf, "Audio-Visual Unit Selection for the Synthesis of Photo-Realistic Talking-Heads," *Proc. Int'l Congress on Math. Education 2000*, vol. 2, pp. 619-622, 2000.
- [40] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, 1989.
- [41] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin, "Real-Time Speech Motion Synthesis from Recorded Motions," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, 2004.
- [42] X. Huang, A. Acero, and X. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall, 2001.

- [43] G. Feng, "Data Smoothing by Cubic Spline Filters," *IEEE Trans. Signal Processing*, vol. 46, no. 10, pp. 2790-2796, 1998.
- [44] B. Wise, R. Cole, S. van Vuuren, S. Schwartz, L. Snyder, N. Ngampatipatpong, J. Tuantranont, and B. Pellom, "Learning to Read with a Virtual Tutor: Foundations Literacy," *Interactive Literacy Education*, C. Kinzer and L. Verhoeven, eds., Mahwah, N.J.: Lawrence Erlbaum, 2005.



**Jiyong Ma** received the PhD degree in computer science from the Harbin Institute of Technology, China, in 1999 and the BS degree in computational mathematics from Helongjiang University in 1984. Prior to joining the Center for Spoken Language Research (CSLR), he was a postdoctoral researcher at the Institute of Computing Technology at the Chinese Academy of Sciences from March 1999 to February 2001. His research interests include computer animation, computer vision, speech and speaker recognition, pattern recognition algorithms, and applied and computational mathematics. He has published more than 60 scientific papers. He is a member of the IEEE.



**Ron Cole** has studied speech recognition by human and machine for the past 35 years, and has published more than 150 articles in scientific journals and published conference proceedings. In 1990, he founded the Center for Spoken Language Understanding (CSLU) at the Oregon Graduate Institute. In 1998, he founded the Center for Spoken Language Research (CSLR) at the University of Colorado, Boulder. He is a member of the IEEE.



**Bryan Pellom** received the BSc degree in computer and electrical engineering from Purdue University, West Lafayette, Indiana, in 1994 and the MSc and PhD degrees in electrical engineering from Duke University in 1996 and 1998, respectively. From 1999 to 2002, he was a research associate with the Center for Spoken Language Research (CSLR), University of Colorado, Boulder. His research activities were focused on automatic speech recognition, concatenative speech synthesis, and spoken dialog systems. Since 2002, he has been a research assistant professor in the Department of Computer Science and with the CSLR. His current research is focused in the area of large vocabulary speech recognition. He is a member of the IEEE.



**Wayne Ward** received the BA degree in 1973 from Rice University, Houston, Texas, with a double major in mathematical science and psychology. He received the MS and PhD degrees in psychology from the University of Colorado, Boulder, in 1981 and 1984, respectively. He is a full time research faculty member in the Center for Spoken Language Research. He works in the area of spoken language processing and dialogue modeling for conversa-

tional computer systems and information retrieval in question/answering systems.



**Barbara Wise** received the BA degree in psychology with honors from Stanford University and the PhD degree in developmental psychology from the University of Colorado in Boulder. Dr. Wise developed the Linguistic Remedies program and classes, where she shares her own and others' research knowledge and teaching ideas with professionals and parents concerned with children with reading difficulties. She has conducted teacher training workshops in many

towns in Colorado, as well as in Texas, California, Connecticut, Pennsylvania, and Washington.

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**