Expressive Facial Animation Synthesis by Learning Speech Coarticulation and Expression Spaces

Zhigang Deng, *Member*, *IEEE Computer Society*,
Ulrich Neumann, *Member*, *IEEE Computer Society*, J.P. Lewis, *Member*, *IEEE Computer Society*,
Tae-Yong Kim, Murtaza Bulut, and Shri Narayanan, *Senior Member*, *IEEE*

Abstract—Synthesizing expressive facial animation is a very challenging topic within the graphics community. In this paper, we present an expressive facial animation synthesis system enabled by automated learning from facial motion capture data. Accurate 3D motions of the markers on the face of a human subject are captured while he/she recites a predesigned corpus, with specific spoken and visual expressions. We present a novel motion capture mining technique that "learns" speech coarticulation models for diphones and triphones from the recorded data. A Phoneme-Independent Expression Eigenspace (PIEES) that encloses the dynamic expression signals is constructed by motion signal processing (phoneme-based time-warping and subtraction) and Principal Component Analysis (PCA) reduction. New expressive facial animations are synthesized as follows: First, the learned coarticulation models are concatenated to synthesize neutral visual speech according to novel speech input, then a texture-synthesis-based approach is used to generate a novel dynamic expression signal from the PIEES model, and finally the synthesized expression signal is blended with the synthesized neutral visual speech to create the final expressive facial animation. Our experiments demonstrate that the system can effectively synthesize realistic expressive facial animation.

Index Terms—Facial animation, expressive speech, animation synthesis, speech coarticulation, texture synthesis, motion capture, data-driven.

1 Introduction

Facial animation is one alternative for enabling natural human computer interaction. Computer facial animation has applications in many fields. For example, in the entertainment industry, realistic virtual humans with facial expressions are increasingly used. In communication applications, interactive talking faces not only make the interaction between users and machines more fun, but also provide a friendly interface and help to attract users [1], [2]. Among the issues concerning the realism of synthesized facial animation, humanlike expression is critical. Despite the need for synthesis of expressive facial animation in these various applications, it still remains a very challenging topic for the computer graphics community. This is because the deformation of a moving face is complex and we humans have an inherent sensitivity to the subtleties of facial motion, but also because human emotion is an extremely difficult interdisciplinary research topic studied by researchers in computer graphics, artificial intelligence, communication, psychology, etc.

In this paper, we present an expressive facial animation synthesis system that learns speech coarticulation models and expression spaces from recorded facial motion capture data. After users specify the input speech (or texts) and its expression type, the system automatically generates corresponding expressive facial animation. The preliminary results of this work have been published in conferences [3], [4].

1.1 System Description

Fig. 1 illustrates the schematic overview of the system. Our system is composed of three stages: recording, modeling, and synthesis. In the recording stage, expressive facial motion and its accompanying audio are recorded simultaneously and preprocessed. In the modeling stage, a new approach is presented to learn speech coarticulation models from facial motion capture data, and a Phoneme-Independent Expression Eigenspace (PIEES) is constructed. In the final synthesis stage, based on the learned speech coarticulation models and the PIEES from the modeling stage, corresponding expressive facial animation is synthesized according to the given input speech/texts and expression.

This synthesis system consists of two subsystems: neutral speech motion synthesis and dynamic expression synthesis. In the speech motion synthesis subsystem, it learns explicit but compact speech coarticulation models from recorded facial motion capture data, based on a weight-decomposition method [5]. Given a new phoneme

- Z. Deng is with the Department of Computer Science, University of Houston, Houston, TX 77004. E-mail: deng@zhigang.org.
- U. Neumann is with the Computer Graphics Lab, 3737 Watt Way, University of Southern California, Los Angeles, CA 90089. E-mail: uneumann@graphics.usc.edu.
- J.P. Lewis is with the Computer Graphics Lab, Stanford University, Gates Building 3B-368, Stanford, CA 94305-9035. E-mail: jplewis@stanford.edu.
- T.-Y. Kim is with Rhythm and Hues Studio, 5404 Jandy Place, Los Angeles, CA 90066. E-mail: tae@rhythm.com.
- M. Bulut and S. Narayanan are with the University of Southern California, 3740 McClintock Avenue, Los Angeles, CA 90089-2564. E-mail: {mbulut, shri}@sipi.usc.edu.

Manuscript received 6 Apr. 2005; revised 20 Dec. 2005; accepted 3 Feb. 2006; published online 8 Sept. 2006.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0049-0405.

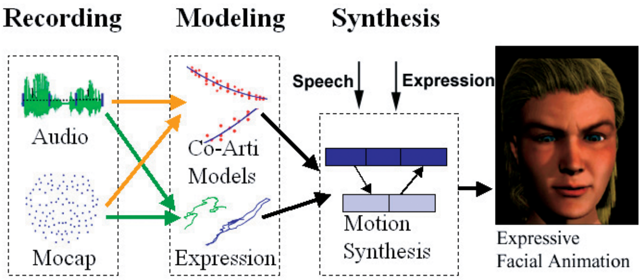


Fig. 1. This figure illustrates the three main stages of our system: recording, modeling, and synthesis. In the recording stage, expressive facial motion and its accompanying audio are recorded simultaneously and preprocessed. In the modeling stage, a new approach is used to learn speech coarticulation models from facial motion capture data, and a Phoneme-Independent Expression Eigenspace (PIEES) is constructed. In the final synthesis stage, based on the learned speech coarticulation models and the PIEES from the modeling stage, expressive facial animation is synthesized according to the given input speech/texts and specified expression.

sequence, this system synthesizes corresponding neutral visual speech motion by concatenating the learned coarticulation models. In the dynamic expression synthesis subsystem, first a Phoneme-Independent Expression Eigenspace (PIEES) is constructed by a phoneme-based time warping and subtraction, and then novel dynamic expression sequences are generated from the constructed PIEES by texture-synthesis approaches [6], [7]. Finally, the synthesized expression signals are weight-blended with the synthesized neutral speech motion to generate expressive facial animation. The compact size of the learned speech coarticulation models and the PIEES make it possible that our system can be used for on-the-fly facial animation synthesis.

To make the concepts used in this work clear, "visual speech animation" (or "lip animation") represents the facial motion only in the lower face region (around the mouth area) and "expressive facial animation" represents expressive facial motion in the full face region (including the upper face region and the lower face region) because motion only in the lower face region is not enough to convey complete emotions. The remainder of the paper is organized as follows: Section 2 reviews previous and related work in facial animation. Section 3 describes the capture and preprocessing of a database of expressive facial motion capture data. Section 4 describes the learning of accurate speech coarticulation models from facial motion capture data. Section 5 details the construction of a phoneme-independent expression eigenspace. Section 6 describes the creation of novel expressive facial animation, given novel input speech/texts and expressions. Section 7 shows the synthesized expressive facial motion results and their evaluation. Finally, conclusions and discussion are given in Section 8.

2 Related Work

In this section, we review related facial animation work. An extensive overview can be found in the well-known facial animation book by Parke and Waters [8].

2.1 Expressive Facial Animation

Cassell et al. [9] present a rule-based automatic system that generates expressions and speech for multiple conversation agents. In their work, the Facial Action Coding Systems (FACS) [10] are used to denote static facial expressions. Noh and Neumann [11] present an "expression cloning" technique to transfer existing expressive facial animation between different 3D face models. This technique and the extended variant by Pyun et al. [12] are very useful for transferring expressive facial motion. However, they are not generative; they cannot be used for generating new expressive facial animation. Chuang et al. [13] learn a facial expression mapping/transformation from training footage using bilinear models and, then, this learned mapping is used to transform input video of neutral talking to expressive talking. In their work, the expressive face frames retain the same timing as the original neutral speech, which does not seem plausible in all cases. Cao et al. [14] present a motion editing technique that applies Independent Component Analysis (ICA) onto recorded expressive facial motion capture data and, then, perform more editing operations on these ICA components, interpreted as expression and speech components separately. This approach is used only for editing existing expressive facial motion, not for the purpose of synthesis. Zhang et al. [15] present a geometry-driven technique for synthesizing expression details on 2D face images. This method is used for static 2D expression synthesis, but the applicability of this method to animate images and 3D face models has not been established. Blanz et al. [16] present an animation technique to reanimate faces in images and video by learning an expression and viseme space from scanned 3D faces. This approach addresses both speech and expressions, but static expression poses do not provide enough information to synthesize realistic dynamic expressive motion. The success of expressive speech motion synthesis by voice puppetry [17] depends heavily on the choice of audio features used and, as pointed out by Brand [17], the optimal audio feature combination for expressive speech motion is still an open problem. Kshirsagar et al. [18], [19] present a PCA-based method for generating expressive speech animation. In their work, static expression configurations are embedded in an expression and viseme space, constructed by PCA. Expressive speech animation is synthesized by weighted blending between expression configurations (corresponding to some points in the expression and viseme space) and speech motion. Additionally, significant effort has been made on expressive virtual characters in complex scenarios [20], [21].

2.2 Speech Animation

The key part of speech animation synthesis is modeling speech coarticulation. In linguistics literature, speech coarticulation is defined as follows: phonemes are not pronounced as an independent sequence of sounds, but, rather, the sound of a particular phoneme is affected by adjacent phonemes. Visual speech coarticulation is analogous. Phoneme-driven methods require animators to design key mouth shapes and, then, empirical smooth functions [22], [23], [24], [25], [26], [38] or coarticulation rules [27], [28], [29] are used to generate speech animation. The Cohen-Massaro coarticulation model [22] controls each viseme shape using a target value and a dominance function, and the weighted sum of dominance values determines final mouth shapes. Recent coarticulation work [23], [25], [26] further improved the Cohen-Massaro coarticulation model. For example, Cosi et al. [25] added a temporal resistance function and a shape function for more general cases, such as fast speaking rates. Goff and Benoit [23] calculated the model parameter value of the Cohen-Massaro model by analyzing parameter trajectories measured from a French speaker. Rule-based coarticulation models [27], [28] leave some visemes undefined based on their coarticulation importance and phoneme contexts. Bevacqua and Pelachaud [29] presented an expressive qualifier, modeled from recorded real motion data, to make expressive speech animation. Physics-based methods [30], [31], [32], [33] drive mouth movement by simulating the facial muscles. Physics-based approaches can achieve synthesis realism, but it is hard to solve the optimal parameter values without considerable computing time and tuning efforts.

Data-driven approaches [1], [17], [34], [35], [36], [37] synthesize new speech animation by concatenating prerecorded motion data or sampling from statistical models learned from real data. Bregler et al. [34] present the “video rewrite” method for synthesizing 2D talking faces given novel speech input, based on the collected “triphone video segments.” Instead of using ad hoc coarticulation models and ignoring dynamics factors in speech, this approach models the coarticulation effect with “triphone video segments,” but it is not generative (i.e., the coarticulation cannot be applied to other faces without retraining). Cosatto [1] and Cao et al. [35] further extend “the triphone combination idea” used in “video rewrite” [34] to longer phoneme segments in order to generate new speech animation. Brand [17] learns an HMM-based facial control model by an entropy minimization learning algorithm from voice and video training data and, then, effectively synthesizes full facial motions from a novel audio track. This approach models coarticulation, using the Viterbi algorithm through vocal HMMs to search for the most likely facial state sequence, which is used for predicting facial configuration sequences. Ezzat et al. [36] learn a multidimensional morphable model from a recorded video database that requires a limited set of mouth image prototypes and use the magnitude of diagonal covariance matrices of phoneme clusters to represent coarticulation effects: The larger covariance of a phoneme cluster means this phoneme has a smaller coarticulation and vice versa. Instead of constructing a phoneme segment database [1], [34], [35], [36], Kshirsagar and Thalmann [37] present a syllable-based approach to synthesize novel speech animation. In their approach, captured facial motions are categorized into syllable motions and, then, new speech animation is achieved by concatenating syllable motions optimally chosen from the syllable motion database. However, most of the above data-driven approaches are restricted to synthesizing neutral speech animation and their applications for expressive speech animation synthesis have not been fully demonstrated yet. Additionally, in the above data-driven approaches, speech coarticulation is expressed as an implicit function (e.g., covariance) of the particular facial data. An explicit coarticulation model that can be applied to other face data has not been developed. Additional in-depth discussions on coarticulation can be found in [22].

2.3 Our Model

The speech coarticulation modeling approach presented in this work constructs explicit and compact speech coarticulation models from real human motion data. Both coarticulation between two phonemes (*diphone coarticulation*) and coarticulation among three phonemes (*triphone coarticulation*) are learned. The presented coarticulation modeling approach offers two advantages: 1) It produces an explicit coarticulation model that can be applied to any face model rather than being restricted to “recombinations” of the original facial data. 2) It naturally bridges data-driven approaches (that accurately model the dynamics of real human speech) and flexible keyframing approaches (preferred by animators).

Our coarticulation modeling approach can be easily extended to model the coarticulation of longer phoneme sequences, e.g., those with more than three phonemes, with the cost of requiring significantly more training data because of the “combinational sampling.” As a reasonable trade-off between training data and output realism, diphone and triphone coarticulation models are used in this work.

The dynamic expression synthesis approach presented in this work shares similarities with [18], [19], but the notable distinction of our approach is that expressions are treated as a dynamic process, not as static poses as in [18], [19]. In general, the expression dynamics include two aspects: 1) *Expressive Motion Dynamics (EMD)*: Even in an invariant level of anger, people seldom keep their eyebrows at the same height for the entire duration of the expression. Generally, expressive motion is a dynamic process, not statically corresponding to some fixed facial configurations; 2) *Expression Intensity Dynamics (EID)*: Both the intensity of human expressions and the type of expression may vary over time, depending on many factors, including speech contexts. Varying blending weights over time in [18], [19] can simulate the EID, but the EMD are not modeled because the same static expressive facial configurations are used. In our approach, the EMD is embodied in the constructed PIEES as continuous curves. The optional expression-intensity control is used for simulating EID, similar to [18], [19].

3 Data Acquisition and Preprocessing

A VICON motion capture system [39] with camera rigs (Fig. 2a) with a 120 Hz sampling rate was used to capture

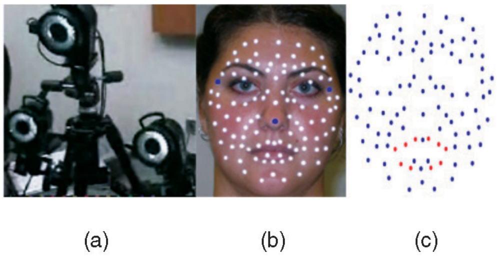


Fig. 2. Illustration of motion data acquisition. (a) The camera rigs of the motion capture system. (b) A snapshot of the captured subject. (c) The markers used in the data acquisition stage. Red points illustrate the markers only used for coarticulation learning, due to occlusions caused by head motion and tracking errors.

the expressive facial motion data of an actress speaking at a normal pace, with markers on her face. The actress was directed to speak a custom phoneme-balanced corpus (about 200 phoneme-rich sentences). The corpus was designed to cover most of the frequently used diphone combinations analyzed from the CMU Sphinx English dictionary. The actress spoke the same corpus four times (each time with different expression). In this work, four basic expressions (neutral, happy, angry, and sad) are considered. The actress was asked to speak the sentences with full intensity expressions. The markers' motion and aligned audio were recorded by the system simultaneously. Fig. 2 illustrates the facial motion capture.

The FESTIVAL system [40] was used to perform phoneme-alignment by aligning each phoneme with its corresponding motion capture segments. This alignment work was done by inputting audio and its accompanying text scripts into the speech recognition program in a forced-alignment mode. Fig. 3 visualizes the phoneme-alignment result of a recorded sentence, "That dress looks like it comes from Asia."

After that, the motion capture data were normalized:

1. All data points were translated in order to force a certain nose point to be the local coordinate center of each frame (Fig. 2b).
2. One frame with neutral and closed-mouth head pose was chosen as a reference frame.
3. Three approximately rigid points (the nose point and corner points of eyes) define a local coordinate system for each frame (Fig. 2b).
4. Each frame was rotated to align it with the reference frame.

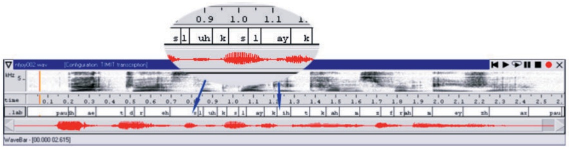


Fig. 3. Illustration of the phoneme-alignment result for a recorded sentence, "That dress looks like it comes from Asia," (using *WaveSurfer* software [41]). Its corresponding phoneme transcript from the FESTIVAL system is "pau-dh-ae-t-d-r-eh-s-l-uh-k-s-l-ay-k-ih-t-k-ah-m-z-f-r-ah-m-ey-zh-ax-pau."

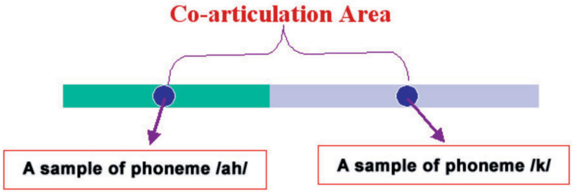


Fig. 4. Illustration of phoneme samples and a coarticulation area for phonemes /ah/ and /k/.

4 Learning Speech Coarticulation

In this section, we detail the construction of explicit speech coarticulation models. Since we specifically focus on speech coarticulation modeling here, only facial motion capture data with a neutral expression are used to learn "speech coarticulation." Due to occlusions caused by head motion and tracking errors, only 10 markers around the mouth area (see red points on Fig. 2c) are used. The dimensionality of these motion vectors (concatenated 10-markers' 3D motion) is reduced using the EM-PCA algorithm [42] because, instead of directly applying Singular Value Decomposition (SVD) on the full data set, the EM-PCA algorithm solves eigenvalues and eigenvectors iteratively using the Expectation-Maximization (EM) algorithm and it uses less memory than regular PCA. The motion data are reduced from the original 30 dimensions to five dimensions, covering 97.5 percent of the variation. In this section, these phoneme-annotated five-dimensional PCA coefficients are used.

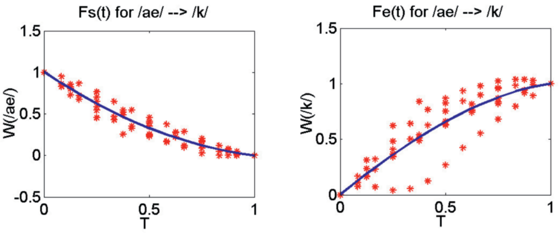
Each phoneme with its duration is associated with a PCA coefficient subsequence. The middle frame of its PCA coefficient subsequence is chosen as a representative sample of that phoneme (termed *a phoneme sample*). In other words, a phoneme sample is a five-dimensional PCA coefficient vector. Hence, the PCA coefficient subsequence between two adjacent phoneme samples captures the coarticulation transition between two phonemes (termed *coarticulation area*). Fig. 4 illustrates phoneme samples and coarticulation area. Then, the weight decomposition method adopted from [5] is used to construct phoneme-weighting functions.

Assume a motion capture segment $M[P_s, P_e]$ for a specific diphone pair $[P_s, P_e]$ is included in the database (see Fig. 4). S_s is the phoneme sample of the starting phoneme (phoneme P_s in this case) at time T_s, and S_e is the phoneme sample of the ending phoneme (phoneme P_e in this case) at time T_e. Notice that subscript s stands for the starting phoneme and subscript e stands for the ending phoneme in the above notations. F_j at time T_j is one intermediate PCA coefficient frame in the coarticulation area of $M[P_s, P_e]$. Equation (1) is solved to get the normalized time t_j ($0 \leq t_j \leq 1$), and (2) is solved, in the least-square sense, to get the weight of the starting phoneme, $W_{j,s}$, and the weight of the ending phoneme, $W_{j,e}$:

$$t_j = (T_j - T_s)/(T_e - T_s), \tag{1}$$

$$F_j = W_{j,s} \times S_s + W_{j,e} \times S_e, \tag{2}$$

where $T_s \leq T_j \leq T_e$, $W_{j,s} \geq 0$ and $W_{j,e} \geq 0$.



(a) (b)

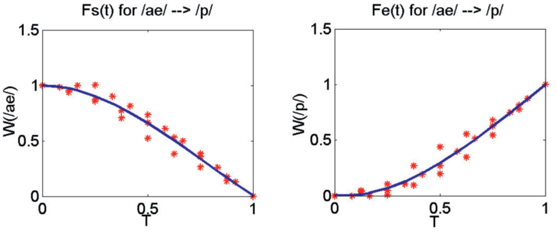
Fig. 5. An example of diphone coarticulation functions (for a phoneme pair /ae/ and /k/: $F_s(t)$ and $F_e(t)$). Each star point in the figure represents one calculated time-weight relation: (a) $< t_j^i, W_{j,s}^i >$ and (b) $< t_j^i, W_{j,e}^i >$.

Thus, we obtain two time-weight relations $< t_j, W_{j,s} >$ and $< t_j, W_{j,e} >$ for any intermediate frame F_j. Assume there are total N motion capture segments for this specific diphone pair $[P_s, P_e]$ in the database and the coarticulaton area of the ith segment has K_i frames. Then, the gathered time-weight relations $< t_j^i, W_{j,s}^i >$ and $< t_j^i, W_{j,e}^i >$ $(1 \leq i \leq N$ and $1 \leq j \leq K_i)$ encode all the coarticulation transitions for the diphone $[P_s, P_e]$ (the superscript of notations denotes which motion capture segment). Two polynomial curves $F_s(t)$ and $F_e(t)$ are used to fit these time-weight relations. Mathematically, we solve $F_s(t)$ and $F_e(t)$ by minimizing the following error functions:

$$e_S(P_s, P_e) = \sum_{i=1}^{N}\sum_{j=1}^{K_i}(F_s(t_j^i) - W_{j,s}^i)^2, \tag{3}$$

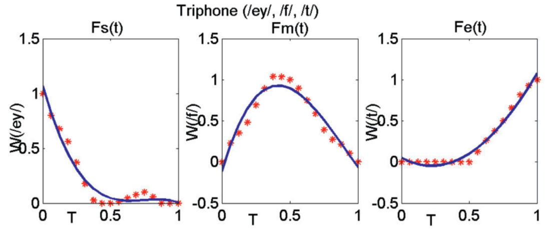
$$e_E(P_s, P_e) = \sum_{i=1}^{N}\sum_{j=1}^{K_i}(F_e(t_j^i) - W_{j,e}^i)^2, \tag{4}$$

where $F_s(t)$ and $F_e(t)$ are referred to as the *Starting-Phoneme Weighting Function* and the *Ending-Phoneme Weighting Function*, respectively. Here, we experimentally constrain $F_s(t)$ and $F_e(t)$ to third degree polynomial curves (see follow-up explanations). Fig. 5 and Fig. 6 illustrate two examples of these diphone coarticulation functions. In Fig. 5 and Fig. 6, the decrease of phoneme /ae/ during the transition from phoneme /ae/ to phoneme /k/ is faster than that of the transition from phoneme /ae/ to phoneme /p/.



(a) (b)

Fig. 6. Another example of diphone coarticulation functions (for a phoneme pair /ae/ and /p/: $F_s(t)$ and $F_e(t)$). Each star point in the figure represents one calculated time-weight relation: (a) $< t_j^i, W_{j,s}^i >$ and (b) $< t_j^i, W_{j,e}^i >$.



(a) (b) (c)

Fig. 7. An example of triphone coarticulation functions. It illustrates triphone coarticulation functions for triphone (/ey/, /f/, and /t/): (a) $F_s(t)$, (b) $F_m(t)$, and (c) $F_e(t)$.

For triphone coarticulations, (5) and (6) are analogously solved to get three time-weight relations $< t_j, W_{j,s} >$, $< t_j, W_{j,m} >$, and $< t_j, W_{j,e} >$:

$$t_j = (T_j - T_s)/(T_e - T_s), \tag{5}$$

$$F_j = W_{j,s} \times S_s + W_{j,m} \times S_m + W_{j,e} \times S_e, \tag{6}$$

where S_s, S_m, and S_e represent the three phoneme samples and the weight values are nonnegative. In a similar way to (3) and (4), three polynomial weighting functions $F_s(t)$, $F_m(t)$, and $F_e(t)$ are used to fit these time-weight relations separately and $e_S(P_s, P_m, P_e)$, $e_M(P_s, P_m, P_e)$, and $e_E(P_s, P_m, P_e)$ are similarly calculated. We termed $F_s(t)$ as the *Starting-Phoneme Weighting Function*, $F_m(t)$ as the *Middle-Phoneme Weighting Function*, and $F_e(t)$ as the *Ending-Phoneme Weighting Function*. Fig. 7 and Fig. 8 illustrate these triphone coarticulation functions for two triphone cases.

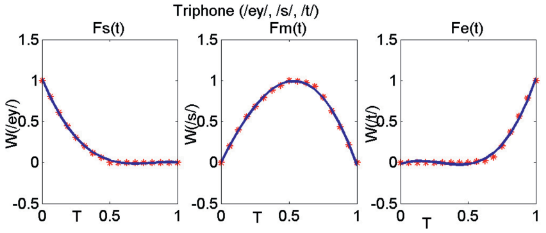
To determine the optimal degree for polynomial fitting, a fitting cost including a model complexity term is minimized. The fitting cost function $C(\lambda)$ is defined as follows ((7), (8), and (9)):

$$C(\lambda) = \sum_{P_i,P_j} diC(P_i, P_j) + \sum_{P_i,P_j,P_k} triC(P_i, P_j, P_k), \tag{7}$$

$$C(P_i, P_j) = \sum_{\theta=S|E} e_\theta(P_i, P_j) + \lambda \sum_{\theta=\{s,e\}} \| F_\theta \|^2, \tag{8}$$

$$C(P_i, P_j, P_k) = \sum_{\theta=S|M|E} e_\theta(P_i, P_j, P_k) + \lambda \sum_{\theta=s|m|e} \| F_\theta \|^2 . \tag{9}$$

Here, λ is the penalty value for model complexity, and $\| F \|^2$ is the sum of function F coefficients' squares. Fig. 9 illustrates the cost curve as a function of the degree



(a) (b) (c)

Fig. 8. Another example of triphone coarticulation functions. It illustrates the coarticulation weighting functions of triphone (/ey/, /s/, and /t/): (a) $F_s(t)$, (b) $F_m(t)$, and (c) $F_e(t)$.

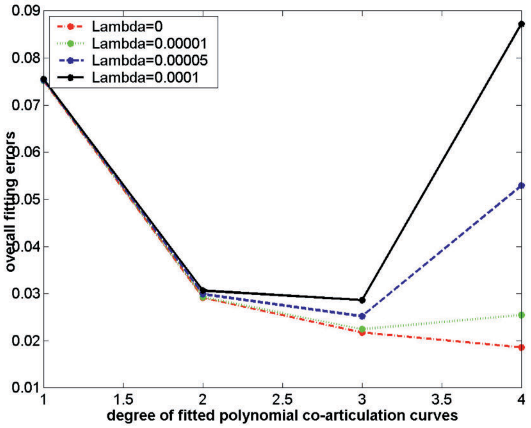


Fig. 9. Fitting error with respect to the degree of fitted coarticulation curves (red is for $\lambda = 0$, green is for $\lambda = 0.00001$, blue is for $\lambda = 0.00005$, and black is for $\lambda = 0.0001$).

of fitting curves. As we can see from Fig. 9, even without the penalty ($\lambda = 0$), n = 3 is still a good trade-off point. In this work, n = 3 is experimentally chosen for fitting polynomial coarticulation curves in this work.

5 Construct Expression Eigenspaces

Since the same sentence material was used for capturing facial motions of the four different expressions and spoken by the subject without different emphasis, the phoneme sequences, except for their timing, are the same. Based on this observation, a phoneme-based time warping and resampling (supersample/down-sample) is applied to the expressive capture data to make them align strictly with neutral data, frame by frame. We should note that the time warping assumption is just an approximation (the velocity/acceleration in the original expressive motion may be impaired in this warping) since expressive speech modulations do involve durational modifications [43]. In this step, eyelid markers are ignored. Fig. 10 and Fig. 11 illustrate this time-warping procedure for a short piece of angry data.

Subtracting neutral motion from aligned expressive motion generates pure expressive motion signals. Since they are strictly phoneme-aligned, we assume that the above subtraction removes "phoneme-dependent" content from expressive speech motion capture data. As such, the extracted pure expressive motion signals are *Phoneme-Independent Expressive Motion Signals (PIEMS)*.

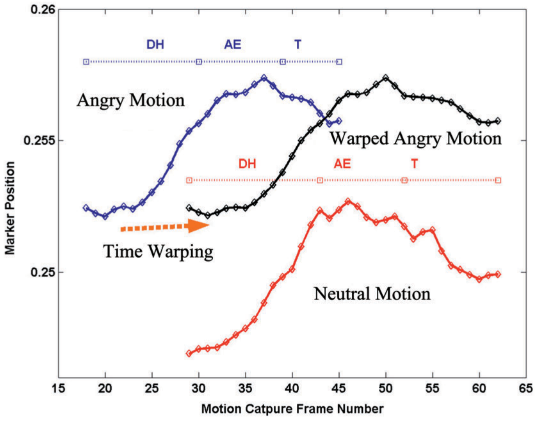


Fig. 10. Illustration of phoneme-based time-warping for the Y position of a particular marker. Although the phoneme timings are different, the warped motion (black) is strictly frame aligned with neutral data (red).

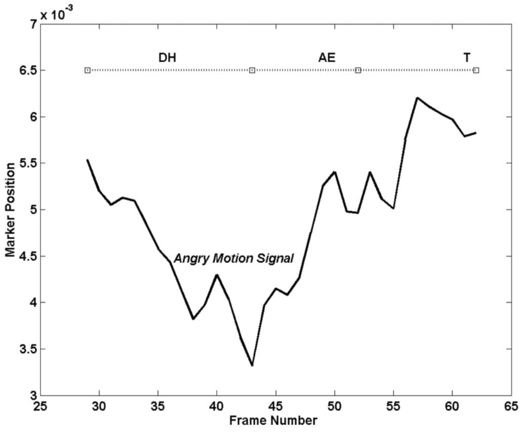


Fig. 11. Extracted phoneme-independent angry motion signal from Fig. 10.

The extracted PIEMS are high dimensional when the 3D motion of all markers are concatenated together. As such, all the PIEMS are put together and reduced to three dimensions, covering 86.5 percent of the variation. The EM-PCA algorithm [42] is used here. In this way, we find a three-dimensional PIEES (Phoneme Independent Expression Eigenspace), where expression is a continuous curve. Fig. 12 and Fig. 13 illustrate the PIEES and the PIEMS. Note that the personality of the captured subject may be irreversibly reflected in the PIEES and only four

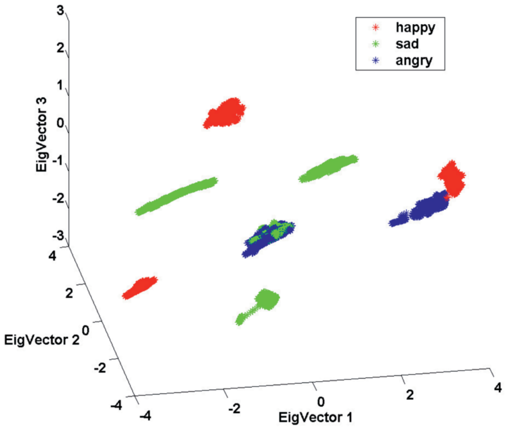


Fig. 12. Plot of three expression signals on the PIEES. It shows that sad signals and angry signals overlap in some places.

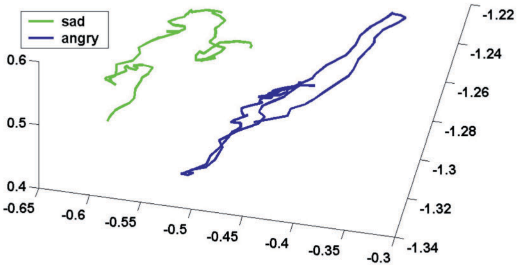


Fig. 13. Plot of two expression sequences in the PIEES. It shows that expression is just a continuous curve in the PIEES.

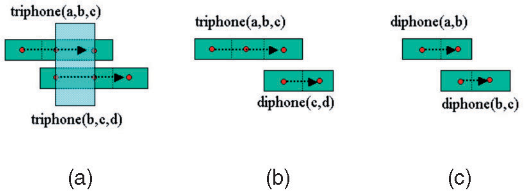


Fig. 14. Illustration of the junctures of adjacent diphones and triphones. The overlapping part (the semitransparent part) in the juncture of two triphones (a) needs to be smoothed. Note that there is another diphone-triphone configuration, similar to (b).

basic expressions are considered. Building person-independent expression eigenspace and modeling the universal expression space are beyond this work.

6 Expressive Facial Animation Synthesis

6.1 Speech Motion Synthesis

After coarticulation models and PIEES are learned, our approach synthesizes new expressive facial animation, given novel phoneme-annotated speech/texts input and key viseme mappings. A total of 13 key viseme shapes (each corresponding to a visually similar group of phonemes, e.g., /p/, /b/, and /m/) are used. The mapped key shapes (key visemes) are 3D facial control point (marker) configurations. New speech animations are synthesized by blending these key shapes by concatenating the learned coarticulation models, and dynamic phoneme-independent expression sequences are synthesized from the constructed PIEES by a texture synthesis approach. Finally, these two are weight-blended to produce expressive facial animation.

Given a phoneme sequence $P_1, P_2, \cdots, P_n$ with timing labels, blending these key shapes generates intermediate animations. The simplest approach would be linear interpolation between adjacent pairs of key shapes, but linear interpolation simply ignores any coarticulation effects. Inspired by [35], a greedy searching algorithm is proposed to concatenate these learned coarticulation models. Fig. 14 illustrates all possible juncture cases for adjacent diphones/triphones. Only Fig. 14a needs motion blending. In this work, the motion blending technique presented in [44] is used. Equation (10) describes the used parametric rational G^n continuous blending functions [44]:

$$b_{n,\mu}(t) = \frac{\mu(1-t)^{n+1}}{\mu(1-t)^{n+1} + (1-\mu)t^{n+1}}, \tag{10}$$

where t is in [0,1], μ is in (0,1), and $n \geq 0$. Algorithm 1 describes the procedure of the speech motion synthesis algorithm. Note that in the case that diphone models for specific diphone combinations are not available (not included in the training data), cosine interpolation is used as an alternative.

Algorithm 1 MotionSynthesis

Input: $P_{1\rightarrow n}, Keys$

Output: $Motion$

```
$i \leftarrow 1$, $prevTriphone \leftarrow FALSE$, $Motion = \phi$
while $i < n$ do
    if $i + 2 \leq n$ and $triM(P_i \rightarrow P_{i+2})$ exists then
        $NewMo = $ synth$(triM(P_i \rightarrow P_{i+2}), Keys)$
        if $preTriphone$ then
            $Motion = $ catBlend$(Motion, NewMo)$
        else
            $Motion = $ concat$(Motion, NewMo)$
        end if
        $preTriphone = $ TRUE, $i = i + 1$
    else
        if $preTriphone$ then
            $preTriphone = $ FALSE
        else
            $NewMo = $ synth$(diM(P_i \rightarrow P_{i+1}), Keys)$
            $Motion = $ concat$(Motion, NewMo)$
        end if
        $i = i + 1$
    end if
end while
```

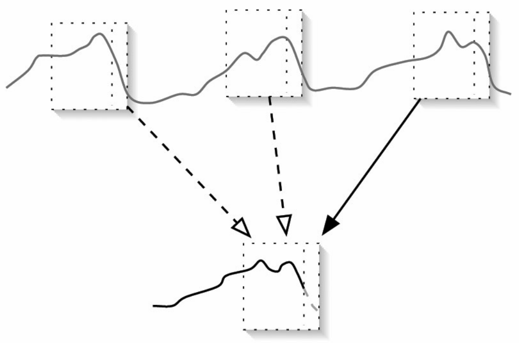


Fig. 15. Illustration of patch-based sampling for expression signal synthesis in this work.

6.2 Expressive Motion Synthesis

On the expression side, from Fig. 13, we observe that expression is just a continuous curve in the low-dimensional PIEES. Texture Synthesis, originally used in 2D image synthesis, is a natural choice for synthesizing novel expression sequences. Here, nonparametric sampling methods [6], [7] are used. The patch-based sampling algorithm [7] is chosen due to its real-time efficiency. Its basic idea is to grow one texture patch (fixed size) at a time, randomly chosen from qualified candidate patches in the input texture sample. In this work, each texture sample (analogous to a pixel in the 2D image texture case) consists of three elements: the three coefficients of the projection of a motion vector on the three-dimensional PIEES. Fig. 15 sketches this synthesis procedure. The parameters of patch-based sampling [7] for this case are patch size = 30, the size of the boundary zone = 5, and the tolerance extent = 0.03.

As mentioned in the data acquisition section (Section 3), the expressive facial motion data used for extracting the PIEES are captured with full expressions. However, in real-world applications, humans usually vary their expression intensity over time. Thus, an optional expression-intensity curve scheme is provided to intuitively simulate the EID. This expression-intensity curve is used to control the

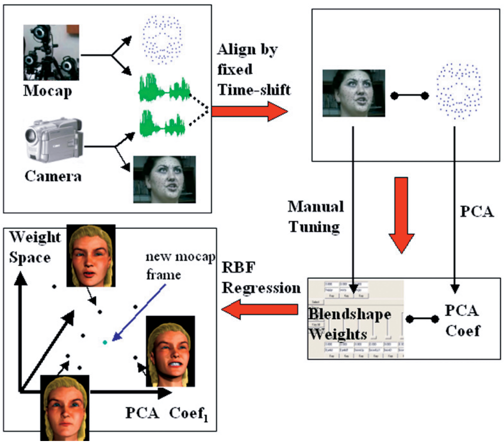


Fig. 16. Schematic overview of mapping marker motions to blendshape weights. It is composed of four stages: data capture stage, creation of mocap-video pairs, creation of mocap-weight pairs, and RBF regression.

weighted-blending of synthesized expression signals and synthesized neutral visual speech. Ideally, the EID (expression intensity curves here) should be automatically extracted from the given audio by emotion-recognition programs [45], [46], [47]. The optional expression-intensity control is a manual alternative to this program.

Basically, an expression intensity curve can be any continuous curve in time versus expression-intensity space, and its range is from 0 to 1, where 0 represents "no expression" (neutral) and 1 represents "full expression." By interactively controlling expression-intensity curves, users can conveniently control expression intensities over time.

6.3 Mapping Marker Motions to 3D Faces

After the marker motion data are synthesized, we need to map them to 3D face models. The target face model is a NURBS face model composed of 46 blendshapes (Fig. 17), such as $\{leftcheekRaise, jawOpen, \cdots\}$. The weight range of each blendshape is $[0, 1]$. A blendshape model B is the weighted sum of some predesigned shape primitives [48]:

$$B = B_0 + \sum_{i=1}^{N} W_i * B_i, \qquad (11)$$

where B_0 is a base face, B_i are delta blendshape bases, and W_i are blendshape weights. In this work, B and B_i (11) are vectors that concatenate all markers' 3D positions.

An RBF-regression-based approach [49] is used to directly map synthesized marker motions to blendshape weights. In the first stage (capture stage), a motion capture system and a video camera are simultaneously used to record the facial motions of a human subject. The audio recordings from the two systems are misaligned with a fixed time-shift because of slight differences in the start time of recording. The manual alignment of these two audio recordings results in strict alignments between mocap frames and video frames (referred to as *mocap-video pairs*). In the second stage, we carefully select a few reference mocap-video pairs that cover the spanned space of visemes and emotions as completely as possible. In the third stage, motion capture data were reduced to a low dimensional space by Principal Component Analysis (PCA). Meanwhile, based on the selected reference video frames (face snapshots), users manually tune the weights of the blendshape face model to perceptually match the model and the reference images, which creates supervised correspondences between the PCA coefficients of motion capture frames and the weights of the blendshape face model (referred to as *mocap-weight pairs*). Taking the reference mocap-weight pairs as training examples, the Radial Basis Function (RBF) regression technique is used to automatically compute blendshape weights for new motion capture frames. Fig. 16 illustrates this process. More details about this mapping algorithm can be found in [49].

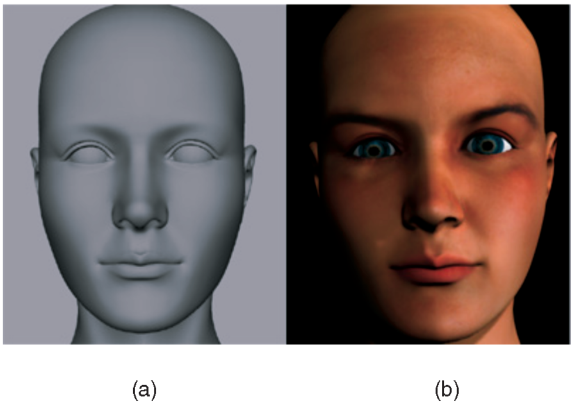

(a) (b)

Fig. 17. Illustration of the used blendshape face model. (a) The smooth shaded model. (b) The rendered model.

In summary, the complete synthesis algorithm can be described in Algorithm 2. Here, procedure *MotionSynthesis* synthesizes neutral visual speech using the above Algorithm 1, procedure *ExprSynthesis* synthesizes novel expression signals with a specified expression from the PIEES, and the procedure *Blend* combines these two together to generate expressive facial motion. Note that this blending is done on the motion marker level. The final procedure *Map2Model* maps the synthesized marker motion to a specific 3D face model.

```
Algorithm 2 ExpressiveFacialAnimationSynthesis
Input: a phoneme sequence with timing $P[1 \ldots N]$
Input: specified key shapes $keyShapes$
Input: specified expression information $Expr$
Input: specified 3D face models $Model$
Output: $AnimFace$
  1: $SpeechSeq = \textbf{MotionSynthesis}(P, keyShapes)$
  2: $ExpSignal = \textbf{ExprSynthesis}(PIEES, Expr)$
  3: $ExprMotion = \textbf{Blend}(SpeechSeq, ExpSignal)$
  4: $AnimFace = \textbf{Map2Model}(Model, ExprMotion)$
```

7 Results and Evaluations

To evaluate the performance of this expressive facial animation synthesis system, we designed two different tests. The first test is to synthesize new expressive visual speech animation given novel audio/text inputs. The second test is used to verify this approach by comparing

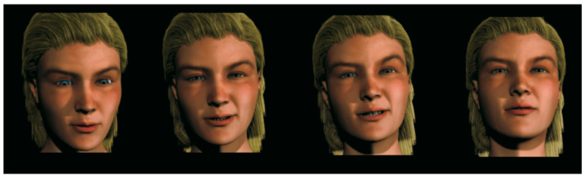


Fig. 18. Some frames of synthesized happy facial animation.

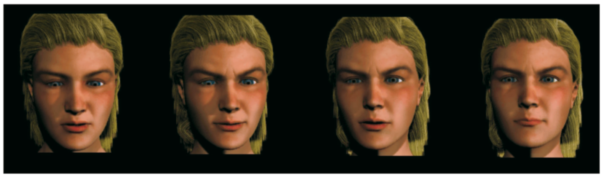


Fig. 19. Some frames of synthesized angry facial animation.

ground-truth motion capture data and synthesized speech motion via trajectory comparisons.

New sentences (not used in the training) and music are used for synthesizing novel speech animation. First, recorded speech (or music), with its accompanying texts (or lyrics), was inputted to the phoneme-alignment program (speech recognition program in a force-alignment mode) to generate a phoneme sequence with timing labels. Then, this phoneme sequence was fed into the facial animation synthesis system to synthesize corresponding expressive facial animation. Fig. 18 and Fig. 19 illustrate some frames of synthesized expressive facial animation.

Modeling eye gaze, blink, teeth, and tongue motion are outside the current scope of this work. For the demonstration video, the automated eye motion method [50] is used to generate eye motion, and the multiresolution hair modeling and rendering system [51] is used. Teeth are assumed to have the same rotation as the jaw with respect to a rotation axis (a straight line below the line between two ears) and jaw rotation angles can be estimated from synthesized mouth motion key points. For tongue motions, instead of using a highly deformable tongue model [52], [26], we simply designed key tongue shapes for phonemes and linear interpolation is used to generate the tongue motion.

We evaluated the learned speech coarticulation models by trajectory comparisons. Several sentences of the original speaker's audio (not used in the previous training stage) were used to synthesize neutral speech animation using this approach. The synthesized motion of the same 10 markers around the mouth area is then compared frame-by-frame with the corresponding ground-truth data. In this evaluation, manually picked key shapes are used that may not perfectly match the motion capture geometry. Fig. 20 shows comparison results for a lower lip marker of one phrase. As we can see from Fig. 20, these trajectories are similar, but the velocity curves have more obvious differences at some places. Its underlying reason could be that, in current work, only markers' 3D positions are used during the modeling stage, while velocity information is ignored. Hence, an interesting future extension could be combining position and velocity for facial animation learning.

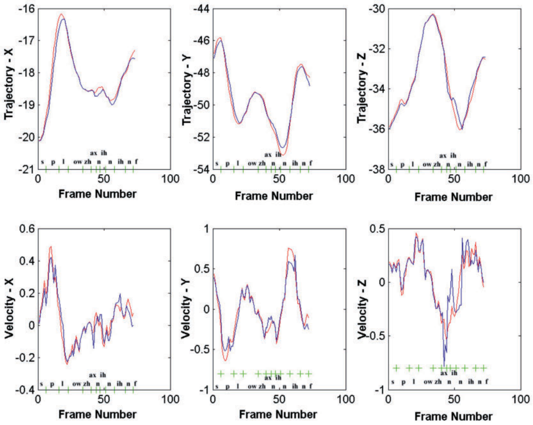


Fig. 20. Comparisons with ground-truth marker motion and synthesized motion. The red line denotes ground-truth motion and the blue denotes synthesized motion. The top row illustrates marker trajectory comparisons and the bottom row illustrates velocity comparisons. Note that the sampling frequency here is 120 Hz. The phrase is "explosion in information technology."

8 Conclusions and Discussion

In this paper, a novel system is presented for synthesizing expressive facial animation. It learns speech coarticulation models from real motion capture data by using a weight-decomposition method and the presented automatic technique for synthesizing dynamic expression models in both the EMD and the EID, improving on previous expression synthesis work [18], [19]. The approach presented in this work learns personality (speech coarticulations and phoneme-independent expression eigenspaces) using data captured from the human subject. The learned personality can then be applied to other target faces. The statistical models learned from real speakers make the synthesized expressive facial animation more natural and lifelike. Our coarticulation modeling approach can be easily extended to model the coarticulation effects among longer phoneme sequences (e.g., five to six-phoneme length) at the cost of requiring significantly more training data.

This system can be used for various applications. For animators, after initial key shapes are provided, this approach can serve as a rapid prototyping tool for generating natural-looking expressive facial animation while simultaneously preserving the expressiveness of the animation. After the system outputs generated facial animation, animators can refine the animation by adjusting the visemes and timing as desired. Because of the very compact size of the coarticulation models and the PIEES learned by this approach, it can be conveniently applied onto mobile-computing platforms (with limited memory), such as PDAs and cell phones.

The coarticulation modeling method presented in this work is efficient and reasonably effective, but it does not differentiate between varying speaking rates. As such, future work could include extending current coarticulation models to handle different speaking rates and affective states; for example, investigating how the learned coarticulation functions (curves) change when speaking rates are increased or decreased. Another major limitation of this coarticulation work is that it depends on "combinational sampling" from training data. Hence, the best results require that the training data have a complete set of phoneme combinations. Additionally, the current system still requires animators to provide key viseme shapes. Future work on automatically extracting key visemes shapes from motion capture data would be a promising way to replace this manual step.

In terms of validating our work, we are aware that objective comparisons are not enough; conducting audio-visual perceptual experiments could be another useful way to evaluate this work, which we plan to pursue in the future. Another consideration is that only 10 markers around the lips do not capture all the details of lip motion; for example, when the lips are closed, inner lips could penetrate each other. We plan to use more markers for facial motion capture in order to further improve and validate this work.

A limitation of the expression synthesis approach in this work is that the interaction between expression and speech is simplified. We assume there is a PIEES extracted by phoneme-based subtraction. The time-warping algorithm used in expression eigenspace construction may cause the loss of velocity/acceleration that is essential to expressive facial motion. We plan to investigate the possibility of learning statistical models for velocity/acceleration patterns in captured expressive speech motion. Transforming these learned patterns back to the synthesized facial motion will further enhance its expressive realism.

A large amount of expressive facial motion data are needed to construct the PIEES, because it is difficult to anticipate in advance how much data are needed to avoid getting "stuck" during the synthesis process. However, after some animation is generated, it is easy to evaluate the variety of synthesized expressions and more data can be obtained if necessary. We plan to look into some automatic ways to avoid "getting stuck" in case the training data are not enough, for example, if the synthesis algorithm cannot find enough qualified candidates with a predefined threshold value, the algorithm should be able to adjust this threshold value adaptively and automatically.

Another limitation of the expression synthesis work is that some visemes may lose their characteristic shapes when blending expression with neutral ones. In future work, we plan to avoid this problem by using "constrained texture synthesis" for expression signal synthesis that imposes certain hard constraints at specified places. Another promising way to avoid the possible relaxation of characteristic shapes is to learn a speech coarticulation model for each affective state.

Most of the learning-based systems face one common difficult concern: what are the optimal parameter decisions/trade-offs involved in the learning system and how do we determine these parameters from the data? Our system has similar issues too. In the part of learning speech coarticulation, we have to make experimental decisions on the fitting degree and the length of learned phoneme sequences (e.g., three for triphones). Additionally, another trade-off between the dimensionality of the learned expression space and the quality of synthesized expressions is concerned in the expression synthesis part. Understanding these trade-offs and their effects on the system performance would be an important and interesting direction to be pursued in the future.

We are aware that expressive eye motion and head motion are critical parts of expressive facial animation since the eye is one of the strongest cues to the mental state of a person and head movement is somehow correlated with speech contents [53], [54]. Simply adding prerecorded head movement and eye motion onto new synthesized talking faces that speak novel sentences may create unrealistic mouth-head gesture coordination. Future work on speech-driven expressive eye motion and head motion synthesis can greatly enhance the realism of synthesized expressive facial animation.

Acknowledgments

This research has been funded by the Integrated Media System Center at the University of Southern California (USC), a US National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the US National Science Foundation. The authors give special thanks to Hiroki Itokazu, Bret St. Clair, Pamela Fox, and Shawn Drost for model preparation, and to Jun-Yong Noh and Douglas Fidaleo for useful discussion. The authors also appreciate many valuable comments from other colleagues in the CGIT Lab at USC. The authors would also like to thank the TVCG editor and anonymous reviewers for their suggestive and insightful comments.

References

[1] E. Cosatto, "Sample-Based Talking-Head Synthesis," PhD thesis, Swiss Federal Inst. of Technology, 2002.

[2] I.S. Pandzic, "Facial Animation Framework for the Web and Mobile Platforms," *Proc. Seventh Int'l Conf. 3D Web Technology,* 2002.

[3] Z. Deng, M. Bulut, U. Neumann, and S.S. Narayanan, "Automatic Dynamic Expression Synthesis for Speech Animation," *Proc. IEEE Conf. Computer Animation and Social Agents (CASA),* pp. 267-274, July 2004.

[4] Z. Deng, J.P. Lewis, and U. Neumann, "Synthesizing Speech Animation by Learning Compact Speech Coarticulation Models," *Proc. Computer Graphics Int'l Conf. (CGI),* pp. 19-25, June, 2005.

[5] K. Pullen and C. Bregler, "Motion Capture Assisted Animation: Texturing and Synthesis," *ACM Trans. Graphics,* pp. 501-508, 2002.

[6] A. Efros and T.K. Leung, "Texture Synthesis by Nonparametric Sampling," *Proc. Int'l Conf. Computer Vision (ICCV '99),* pp. 1033-1038, 1999.

[7] L. Liang, C. Liu, Y.Q. Xu, B. Guo, and H.Y. Shum, "Real-Time Texture Synthesis by Patch-Based Sampling," *ACM Trans. Graphics,* vol. 20, no. 3, 2001.

[8] F.I. Parke and K. Waters, *Computer Facial Animation.* Wellesley, Mass.: AK Peters, 1996.

[9] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone, "Animated Conversation: Rule-Based Generation of Facial Expression, Gesture and Spoken Intonation for Multiple Conversational Agents," *Proc. ACM SIGGRAPH Conf.,* pp. 413-420, 1994.

[10] P. Ekman and W.V. Friesen, *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues.* Prentice-Hall, 1975.

[11] J.Y. Noh and U. Neumann, "Expression Cloning," *Proc. ACM SIGGRAPH Conf.,* pp. 277-288, 2001.

[12] H. Pyun, Y. Kim, W. Chae, H.W. Kang, and S.Y. Shin, "An Example-Based Approach for Facial Expression Cloning," *Proc. 2003 ACM SIGGRAPH/Eurographics Symp. Computer Animation,* pp. 167-176, 2003.

[13] E.S. Chuang, H. Deshpande, and C. Bregler, "Facial Expression Space Learning," *Proc. Pacific Graphics Conf.,* pp. 68-76, 2002.

[14] Y. Cao, P. Faloutsos, and F. Pighin, "Unsupervised Learning for Speech Motion Editing," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation,* 2003.

[15] Q. Zhang, Z. Liu, B. Guo, and H. Shum, "Geometry-Driven Photorealistic Facial Expression Synthesis," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation,* 2003.

[16] V. Blanz, C. Basso, T. Poggio, and T. Vetter, "Reanimating Faces in Images and Video," *Computer Graphics Forum (Proc. Eurographics 2003 Conf.),* vol. 22, no. 3, 2003.

[17] M. Brand, "Voice Puppetry," *Proc. ACM SIGGRAPH Conf.,* pp. 21-28, 1999.

[18] S. Kshirsagar, T. Molet, and N.M. Thalmann, "Principal Components of Expressive Speech Animation," *Proc. Computer Graphics Int'l Conf.,* 2001.

[19] A.S. Meyer, S. Garchery, G. Sannier, and N.M. Thalmann, "Synthetic Faces: Analysis and Applications," *Int'l J. Imaging Systems and Technology,* vol. 13, no. 1, pp. 65-73, 2003.

[20] C. Pelachaud, N. Badler, and M. Steedman, "Generating Facial Expressions for Speech," *Cognitive Science,* vol. 20, no. 1, pp. 1-46, 1994.

[21] C. Pelachaud and I. Poggi, "Subtleties of Facial Expressions in Embodied Agents," *J. Visualization and Computer Animation,* vol. 13, no. 5, pp. 301-312, 2002.

[22] M.M. Cohen and D.W. Massaro, "Modeling Coarticulation in Synthetic Visual Speech," *Models and Techniques in Computer Animation,* pp. 139-156, 1993.

[23] B.L. Goff and C. Benoit, "A Text-to-Audiovisual-Speech Synthesizer for French," *Proc. Int'l Conf. Spoken Language Processing (ICSLP),* pp. 2163-2166, 1996.

[24] G. Kalberer and L.V. Gool, "Face Animation Based on Observed 3D Speech Dynamics," *Proc. IEEE Computer Animation Conf.,* pp. 20-27, 2001.

[25] P. Cosi, C.E. Magno, G. Perlin, and C. Zmarich, "Labial Coarticulation Modeling for Realistic Facial Animation," *Proc. Int'l Conf. Multimodal Interfaces,* pp. 505-510, 2002.

[26] S.A. King and R.E. Parent, "Creating Speech-Synchronized Animation," *IEEE Trans. Visualization and Computer Graphics,* vol. 11, no. 3, pp. 341-352, 2005.

[27] C. Pelachaud, "Communication and Coarticulation in Facial Animation," PhD thesis, Univ. of Pennsylvania, 1991.

[28] J. Beskow, "Rule-Based Visual Speech Synthesis," *Proc. Eurospeech Conf.,* 1995.

[29] E. Bevacqua and C. Pelachaud, "Expressive Audio-Visual Speech," *J. Visualization and Computer Animation,* vol. 15, nos. 3-4, pp. 297-304, 2004.

[30] 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.

[31] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic Modeling for Facial Animation," *Proc. ACM SIGGRAPH Conf.,* pp. 55-62, 1995.

[32] K. Waters and J. Frisble, "A Coordinated Muscle Model for Speech Animation," *Proc. Graphics Interface Conf.,* pp. 163-170, 1995.

[33] K. Kähler, J. Haber, and H.P. Seidel, "Geometry-Based Muscle Modeling for Facial Animation," *Proc. Graphics Interface Conf.,* 2001.

[34] C. Bregler, M. Covell, and M. Slaney, "Video Rewrite: Driving Visual Speech with Audio," *Proc. ACM SIGGRAPH Conf.,* pp. 353-360, 1997.

[35] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin, "Real-Time Speech Motion Synthesis from Recorded Motions," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation,* pp. 345-353, 2004.

[36] T. Ezzat, G. Geiger, and T. Poggio, "Trainable Videorealistic Speech Animation," *ACM Trans. Graphics,* vol. 21, no. 3, pp. 388-398, 2002.

[37] S. Kshirsagar and N.M. Thalmann, "Visyllable Based Speech Animation," *Computer Graphics Forum (Proc. Eurographics Conf.),* vol. 22, no. 3, 2003.

[38] J.P. Lewis, "Automated Lip-Sync: Background and Techniques," *J. Visualization and Computer Animation,* pp. 118-122, 1991.

[39] B. Bodenheimer, C.F. Rose, S. Rosenthal, and J. Pella, "The Process of Motion Capture: Dealing with the Data," *Proc. Eurographics Workshop Computer Animation and Simulation,* 1997.

[40] Festival Speech Synthesis System, http://www.cstr.ed.ac.uk/projects/festival, 2004.

[41] WaveSurfer, http://www.speech.kth.se/wavesurfer, 2005.

[42] S. Roweis, "EM Algorithms for PCA and SPCA," *Proc. Conf. Neural Information Processing Systems (NIPS),* pp. 137-148, 1997.

[43] S. Yildirim, M. Bulut, C.M. Lee, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, and S. Narayanan, "An Acoustic Study of Emotions Expressed in Speech," *Proc. Int'l Conf. Spoken Language Processsing,* 2004.

[44] J. Ma, R. Cole, B. Pellom, W. Ward, and B. Wise, "Accurate Automatic Visible Speech Synthesis of Arbitrary 3D Model Based on Concatenation of Diviseme Motion Capture Data," *Computer Animation and Virtual Worlds,* vol. 15, pp. 1-17, 2004.

[45] R. Cowie, E.D. Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellens, and J.G. Taylor, "Emotion Recognition in Human-Computer Interaction," *IEEE Signal Processing Magazine,* vol. 18, no. 1, pp. 32-80, 2001.

[46] V. Petrushin, "Emotion in Speech: Recognition and Application to Call Centers," *Artificial Neural Networks in Eng.,* pp. 7-10, 1999.

[47] C.M. Lee, S. Narayanan, and R. Pieraccini, "Recognition of Negative Motions from the Speech Signal," *Proc. Conf. Automatic Speech Recognition and Understanding,* 2003.

[48] J.P. Lewis, J. Mooser, Z. Deng, and U. Neumann, "Reducing Blendshape Interference by Selected Motion Attenuation," *Proc. ACM SIGGRAPH Symp. Interactive 3D Graphics and Games (I3D),* pp. 25-29, 2005.

[49] Z. Deng, P. Chiang, P. Fox, and U. Neumann, "Animating Blendshape Faces by Cross-Mapping Motion Capture Data," *Proc. ACM SIGGRAPH Symp. Interactive 3D Graphics and Games,* pp. 43-48, 2006.

[50] Z. Deng, J.P. Lewis, and U. Neumann, "Automated Eye Motion Synthesis Using Texture Synthesis," *IEEE Computer Graphics and Applications,* pp. 24-30, Mar./Apr. 2005.

[51] T.Y. Kim and U. Neumann, "Interactive Multiresolution Hair Modeling and Editing," *ACM Trans. Graphics,* vol. 21, no. 3, pp. 620-629, 2002.

[52] S.A. King and R.E. Parent, "A 3D Parametric Tongue Model for Animated Speech," *J. Visualization and Computer Animation,* vol. 12, no. 3, pp. 107-115, 1990.

[53] H.P. Graf, E. Cosatto, V. Strom, and F.J. Huang, "Visual Prosody: Facial Movements Accompanying Speech," *Proc. IEEE Int'l Conf. Automatic Faces and Gesture Recognition,* 2002.

[54] C. Busso, Z. Deng, U. Neumann, and S. Narayanan, "Natural Head Motion Synthesis Driven by Acoustic Prosody Features," *Computer Animation and Virtual Worlds,* vol 16, nos. 3/4, pp. 283-290, 2005.

Zhigang Deng received the BS degree in mathematics from Xiamen University in 1997, the MS degree in computer science from Peking University 2000, and the PhD degree in computer science from the University of Southern California in 2006. He is an assistant professor in the Department of Computer Science at the University of Houston. His research interests include computer graphics, computer animation, human computer interation, and visualization. He is a member of ACM, ACM SIGGRAPH, and the IEEE Computer Society.

Ulrich Neumann received the MSEE degree from the State University of New York at Buffalo in 1980 and the PhD degree in computer science from the University of North Carolina at Chapel Hill in 1993. He is an associate professor of computer science with a joint appointment in electrical engineering at the University of Southern California (USC). He directs the Computer Graphics and Immersive Technologies (CGIT) Laboratory at USC and is the associate director for research at the US National Science Foundation's Integrated Media Systems Center (IMSC). His current research relates to augmented virtual environments, 3D modeling, and virtual humans. He is a member of the IEEE Computer Society.

J.P. Lewis is a research associate in the graphics lab at Stanford University, California. His interests include facial animation and tracking. He is a member of the IEEE Computer Society.

Tae-Yong Kim received the BS degree in computer engineering from the Seoul National University, the master's degree in computer science from the University of Southern California, and the PhD degree in computer science from the University of Southern California, where he conducted research on hair modeling and rendering techniques. He is currently a research scientist at the Rhythm and Hues Studios. His primary responsibility at R&H includes the development of simulation tools for cloth, hair, and other types of natural phenomena. At R&H, he has developed new simulation techniques used for such movie productions as *The Chronicles of Narnia*, *Garfield*, and *X-Men 2*. His thesis work was published in a SIGGRAPH 2002 paper and many other venues. Recently, he has been a lecturer at SIGGRAPH courses in 2003 and 2004.

Murtaza Bulut received the BS and MS degrees in electrical engineering, from Bilkent University, Ankara, in 2000 and from the University of Southern California (USC), Los Angeles, in 2002, respectively. Currently, he is pursuing the PhD degree in electrical engineering at USC, where he is a research assistant in the Speech Analysis and Interpretation Laboratory (SAIL). His research interests are in digital signal processing, speech analysis and synthesis, and image processing.

Shrikanth Narayanan received the PhD degree from the University of California Los Angeles in 1995. He was with AT&T Research (originally AT&T Bell Labs), first as a senior member and later as a principal member of its technical staff from 1995 to 2000. Currently, he is a professor of electrical engineering, linguistics, and computer science at the University of Southern California (USC). He is a member of the Signal and Image Processing Institute and a research area director of the Integrated Media Systems Center, a US National Science Foundation (NSF) engineering research center, at USC. He was an associate editor of the *IEEE Transactions on Speech and Audio Processing* (2000-2004) and currently serves on the Speech Processing and Multimedia Signal Processing technical committee of the IEEE Signal Processing Society and the Speech Communication committee of the Acoustical Society of America. He is a fellow of the Acoustical Society of America, a senior member of the IEEE, and a member of Tau-Beta-Pi, Phi Kappa Phi, and Eta-Kappa-Nu. He is a recipient of an NSF Career award, a USC Engineering Junior Research Award, a USC Electrical Engineering Northrop Grumman Research Award, a Provost fellowship from the USC Center for Interdisciplinary research, and a corecipient of a 2005 best paper award from the IEEE Signal Processing Society. His research interests are in signals and systems modeling with applications to speech, language, multimodal and biomedical problems. He has published more than 175 papers and has 10 granted/pending US patents.