Synthesizing Speech Animation By Learning Compact Speech Co-Articulation Models

Zhigang Deng* J.P. Lewis† Ulrich Neumann‡

Computer Graphics and Immersive Technologies Lab
Department of Computer Science, Integrated Media System Center
University of Southern California

Figure 1: Some synthesized frames of singing a part of "hunter" music (by dido).

Abstract

While speech animation fundamentally consists of a sequence of phonemes over time, sophisticated animation requires smooth interpolation and co-articulation effects, where the preceding and following phonemes influence the shape of a phoneme. Co-articulation has been approached in speech animation research in several ways, most often by simply smoothing the mouth geometry motion over time. Data-driven approaches tend to generate realistic speech animation, but they need to store a large facial motion database, which is not feasible for real time gaming and interactive applications on platforms such as PDAs and cell phones. In this paper we show that accurate speech co-articulation model with compact size can be learned from facial motion capture data. An initial phoneme sequence is generated automatically from Text-To-Speech (TTS) systems. Then, our learned co-articulation model is applied to the resulting phoneme sequence, producing natural and detailed motion. The contribution of this work is that speech co-articulation models "learned" from real human motion data can be used to generate natural-looking speech motion while simultaneously preserving the expressiveness of the animation via keyframing control. Simultaneously, this approach can be effectively applied to interactive applications due to its compact size.

CR Categories: I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism—Animation; I.2.6 [Artificial Intelligence]: Learning—Knowledge acquisition; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: Speech Animation, Speech Co-Articulation, Facial Animation, Motion Capture, Keyframing Control, Data-Driven, Dynamic Programming

*e-mail: zdeng@graphics.usc.edu
†e-mail: noisebrain@graphics.usc.edu
‡e-mail: uneumann@graphics.usc.edu

1 Introduction

Because of the complexity of deformation of a moving face and our inherent sensitivity to the subtleties of facial animation, creating realistic 3D talking faces remains a challenge problem for computer graphics community. Although several successful automated approaches to computer speech animation have been developed, most commercial speech animation is manually animated. The reasons for this are varied, but among them is the important factor than manual animation provides full control over the result, including the ability to produce cartoon effects such as speech animation in "The Jetsons", and the high efficiency of keyframe interpolations. On the other hand, data-driven approaches [7, 6, 17, 25] tend to produce more realistic animation, but generally they do not provide enough control for animators, so animators cannot use them without considerable efforts.

Most of data-driven approaches need to store a large facial motion database and cannot synthesize speech animation with high efficiency, which prevents the use of these techniques in real-time applications. The time-consuming manual processing stages that are required by most existing data-driven approaches further limit their real time and interactive applications.

In this paper a novel approach is presented to **bridge the data-driven approaches and keyframing approaches**. This approach learns explicit co-articulation models with very compact size, representing the dynamics of speech animation by "mining" facial motion capture data. At the same time, animators can design exaggerated and expressive key shapes as desired. The key shapes can be 3D face models, or other representations such as Facial Action Coding System (FACS) parameters [15]. Given novel phoneme annotated input, such as speech-recognized audio or phoneme sequences from Text-To-Speech (TTS) systems, our approach synthesizes corresponding speech animation in an optimal way, searching for the "optimal co-articulation combinations" from the existing co-articulation models using a dynamic programming technique.

This approach can be used for many scenarios. Animators can use our approach to adjust/design the visemes and timing as desired, to improve quality or produce stylized cartoon effects. This approach can also be used to produce automated real-time speech animation for interactive applications, after initial key shapes are setup.

2 Previous and Related Work

An overview of various facial animation techniques can be found in [31]. Some of the more recent research in this area includes [15, 16, 7, 2, 12, 20, 29, 34, 1, 6, 4, 30, 23, 22, 10, 9, 26, 38, 21, 17, 3, 25, 13, 14]. A number of techniques have also been presented specifically for synthesizing speech animation. Phoneme-driven methods [33, 11, 19, 23] require animators to design key mouth shapes, and then hand-generated smooth functions [11, 19, 23] or co-articulation rules [33] are used to generate speech animation. In physics-based methods [36, 27, 37, 22], the laws of physics drive mouth movement; for example, Lee et al. [27] use three-layer mass-spring muscle and skin structures to generate realistic facial animations. Bregler et al. [7] describe the "video rewrite" method for synthesizing 2D talking faces. Ezzat et al. [17] present a Multidimensional Morphable Model (MMM) method that requires a limited set of mouth image prototypes and effectively synthesizes new speech animation corresponding to novel audio input. Instead of constructing a phoneme database [7, 17], Kshirsagar and Thalmann [25] present a syllable-motion database-based approach to synthesize novel speech animation. In their approach, phoneme sequences are further segmented into syllable sequences, and then syllable motion is chosen from captured syllable motion database and concatenated together. Brand [6] learns a HMM-based facial control model by entropy minimization from example voice and video data and then effectively generates full facial motions from new audio track.

The mouth shape corresponding to a particular phoneme changes slightly depending on the preceding and following phonemes; this is termed *co-articulation*. Traditionally, co-articulation has been approached by simply smoothing the facial parameters or geometry over time [28, 11]. Hand-generated empirical co-articulation models have also been used [19, 23]. Instead of using ad hoc smoothing methods and ignoring dynamics factors, Bregler et al. [7] model the co-articulation effect with "triphone video segments", but it is not generative (i.e. the co-articulation cannot be applied to other faces without retraining). Brand [6] models co-articulation, using the Viterbi algorithm through vocal HMM to search for most likely facial state sequence that further be used for predicting facial configuration sequences. Ezzat et al. [17] use the magnitude of diagonal covariance matrixes of phoneme clusters to represent co-articulation effects: large covariance means small co-articulation, and vice versa. In most of the data-driven approaches above the co-articulation is expressed as an implicit function (e.g. covariance) of the particular facial data. An explicit co-articulation model that can be applied to other face data has not been developed.

The goal of this work is to construct explicit human-like co-articulation models that model the dynamics of real speech accurately and can be used by animators in a controllable way.

Motivated by this goal, in this paper we present a motion capture mining technique for speech co-articulation that constructs explicit, simple, and accurate co-articulation models from real human data. This approach learns optimal polynomial co-articulation curves from the facial motion capture data. Both co-articulation between two phonemes (*diphone co-articulation*) and co-articulation among three phonemes (*triphone co-articulation*) are learned from the real data. Given phoneme-annotated input (audio or text) and viseme shapes, a dynamic programming algorithm is used to search for optimal sequences from existing "diphone co-articulation" and "triphone co-articulation" models and other alternatives.

This approach can be stretched to model the co-articulation among more than three phonemes, and it will need more training data because of the "combinational sampling". As a reasonable trade-off, diphone and triphone co-articulation model are used in this work.The advantages of this approach include:

- It produces an explicit co-articulation model that can be applied to any face model rather than being restricted to "recombinations" of the original facial data.
- It naturally bridges data-driven approaches (that accurately model the dynamics of real human speech) and flexible keyframing approaches (preferred by animators), by combining the realism of data-driven approaches and the controllability of keyframing approaches.

3 System Overview

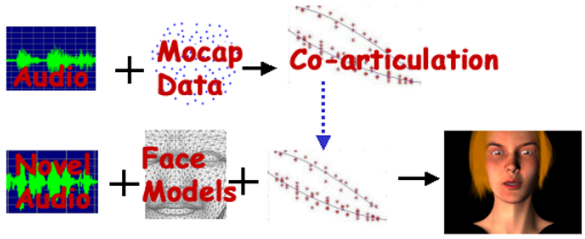

Figure 2: The schematic overview of the speech animation system. The top line illustrates the co-articulation modeling stage, and the bottom line illustrates the speech animation synthesis stage.

Figure 2 gives the overview of this speech animation system. In the preprocessing stage, motion capture data are normalized and aligned with the audio. Then, in the co-articulation modeling stage (top line in Figure 2), various co-articulation curves are learned from the motion capture data. In the synthesis stage (bottom line in Figure 2), given phoneme-annotated input and key shapes designed by animators, corresponding speech animations are synthesized with human-like co-articulation.

The remaining parts are organized as follows: Section 4 describes data gathering and preprocessing. Section 5 describes how to construct diphone and triphone co-articulation models from data. Section 6 describes speech animation synthesis by dynamic programming. The final part describes results and evaluation (Section 7) and discussion and conclusions (Section 8).

4 Data Capture and Preprocessing

We capture facial motion data of a human subject speaking in a normal speed, with markers on the face and recorded corresponding audio and video tracks [5]. The recorded corpus that we created is composed of hundreds of sentences. First, the Festival system [18] is used to align each phoneme with its corresponding motion capture segments. Second, the motion capture data are normalized: 1) all data points are translated in order to make a nose point be the local coordinate center of each frame, 2) one frame with neutral and closed-mouth head pose is 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, and 4) each frame is rotated to align it with the reference frame.

Only the 10 points around the mouth area are used for the speech co-articulation mining (Figure 3). The dimensionality of these motion vectors is reduced using EM-PCA algorithm [35], because EM-PCA can extract eigen-values and eigenvectors from large collections of high dimensional data more efficiently than regular PCA. Instead of directly applying Singular Value Decomposition (SVD) on the full data set, EM-PCA solves eigen-values and eigen-vectors iteratively using the Expectation-Maximization (EM) algorithm.

The motion data are reduced from the original 30 dimensions to 5 dimensions, covering 97.5% of the variation. In the remaining sections, these phoneme-annotated five-dimensional PCA coefficient spaces will be used.

Figure 3: Illustration of ten marker used for this work.

5 Co-Articulation Models

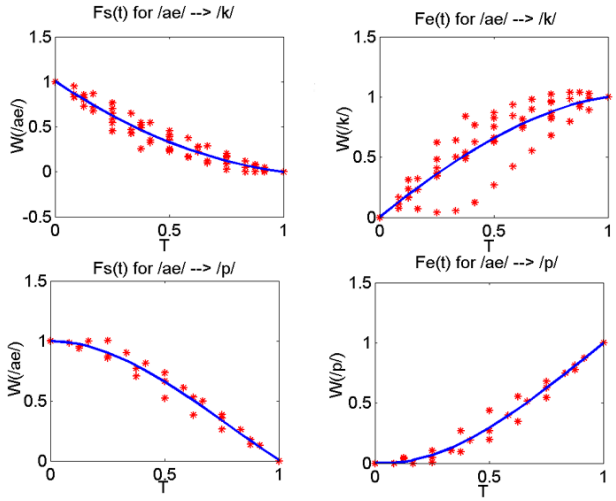

Figure 4: Two examples of diphone co-articulation functions (the top is the co-articulation weighting functions of phoneme pair /ae/ and /k/: Fs(t) and Fe(t) , the bottom is the co-articulation weighting functions of phoneme pair /ae/ and /p/.

In the preprocessing stage described above, each phoneme with its associated duration is associated with a PCA coefficient subsequence. The median of the PCA coefficient subsequence (the middle frame of the sequence) is chosen as a representative sample of that phoneme (a *"phoneme sample"*). So, the PCA coefficient subsequence between two adjacent phoneme samples captures the co-articulation transition between two phonemes. Then, using the weight decomposition method used in [8] to construct phoneme-weighting functions; this approach learns the co-articulation models from the training data. Suppose P_s is the first (starting) phoneme sample at time T_s, P_e is the second (ending) phoneme sample at time T_e,and $P_i (i \geq 1 \cap i \leq k)$ is any intermediate PCA coefficient between them at time $T_i (T_i \geq T_s \cap T_i \leq T_e)$ (its relative time is $t_i = (T_i - T_s)/(T_e - T_s)$). In a least-square sense, the following equation is solved to get two time-weight relations $< t_i, W_{i,0} >$ and $< t_i, W_{i,1} >$:

$$P_i = W_{i,0} \times P_s + W_{i,1} \times P_e \quad (1)$$

where $W_{i,0} \geq 0$ and $W_{i,1} \geq 0$. Gathering all time-weight relations between a specific pair of phonemes, two time-weight sequences ($< t_1, W_{1,0} >, < t_2, W_{2,0} >, \cdots, < t_n, W_{n,0} >$ and $< t_1, W_{1,1} >, < t_2, W_{2,1} >, \cdots, < t_n, W_{n,1} >$) are collected, and two third degree polynomial curves $F_s(t)$ and $F_e(t)$ are used to fit these two time-weight sequences separately (why third degree polynomial curves are chosen will be described in the follow-up). We term $F_s(t)$ and $F_e(t)$ the *Starting-Phoneme Weighting function* and the *Ending-Phoneme Weighting function* respectively. Figure 4 illustrates two examples of these diphone co-articulation functions.

As illustrated in Figure 4, the decay rate rate of phoneme /ae/ during the transition from phoneme /ae/ to phoneme /p/ is faster than that during the transition from phoneme /ae/ to phoneme /k/. The residual fitting error (also termed *diphone co-articulation error*) $E_D^{i,j}$ for the diphone $PH_i \rightarrow PH_j$ is calculated with equation 2, where n is the number of fitted points and PH_i and PH_j are phonemes.

$$E_D^{i,j} = E_s^{i,j} + E_e^{i,j} \quad (2)$$

$$E_s^{i,j} = (\sum_{i=1}^{n} (W_{i,0} - F_s(t_i))^2)/n \quad (3)$$

$$E_e^{i,j} = (\sum_{i=1}^{n} (W_{i,1} - F_e(t_i))^2)/n \quad (4)$$

For the triphone co-articulations, the equation 5 is solved to get three time-weight relations $< t_i, W_{i,0} >$, $< t_i, W_{i,1} >$, and $< t_i, W_{i,2} >$:

$$P_i = W_{i,0} \times P_s + W_{i,1} \times P_m + W_{i,2} \times P_e \quad (5)$$

where P_s, P_m, and P_e represent the three phoneme samples and the weight values are non-negative. Analogously, these time-weight relations are used to fit three weighting functions separately (termed as the *Starting-Phoneme weighting function* $F_s(t)$, *the Middle Phoneme Weighting Function* $F_m(t)$, and the *Ending-Phoneme Weighting Function* $F_e(t)$). The residual fitting error (also termed *triphone co-articulation error*) $E_T^{i,j,k}$ is defined analogously to $E_D^{i,j}$. Figure 5 illustrates these tri-phones co-articulation functions.

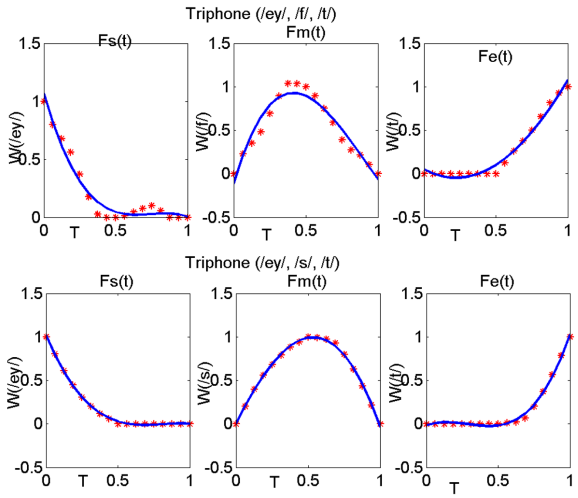

Figure 5: Two examples of triphone co-articulation curves. The top is the co-articulation weighting functions of triphone (/ef/, /f/ and /t/): Fs(t), Fm(t), and Fe(t) , the bottom is the weighting functions of triphone(/ey/, /s/, and /t/).

To determine the appropriate degree for fitting the polynomial curves, we experimentally choose it by checking a total fitting error curve (see Figure 6). The total fitting error C(n) is defined as

follows:

$$C(n) = \sum_{i,j,k} (E_D^{i,j} + E_T^{i,j,k}) \quad (6)$$

Here n is the degree of fitted polynomial curves. Hence, n=3 is experimentally chosen for fitting polynomial co-articulation curves.

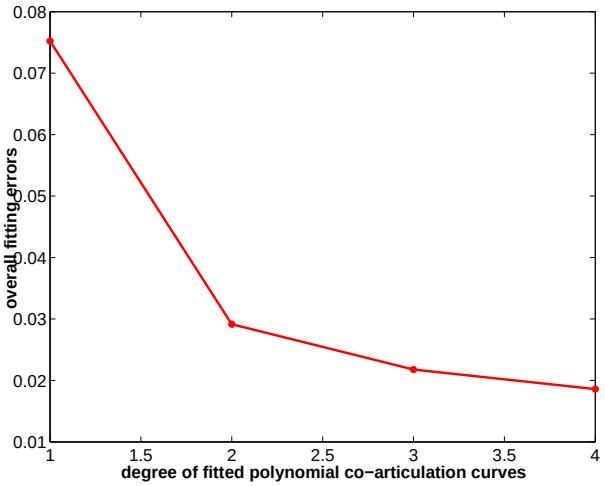

Figure 6: Total fitting error curves vs the degree of fitted co-articulation curves. Here n=3 is experimentally chosen.

In this stage, linear error (to make the names consistent, it is also termed "linear co-articulation error") is also determined. Instead of fitting the third degree polynomial curves, linear fitting (a straight line) is used to fit all the diphones, and we average these linear fitting errors (Eq. 7).

$$LinearErr = \frac{1}{N} \sum_{i,j} E_L^{i,j} \quad (7)$$

Here $E_L^{i,j}$ is defined analogously (Eq. 2) and N is total number of diphones. This average linear co-articulation error is used in Section 6, only in case of corresponding diphone or triphone models are not available.

6 Synthesis with Dynamic Programming

After co-articulation models are learned, this approach synthesizes new speech animation corresponding to given novel phoneme-annotated input and key viseme mapping. The mapped key viseme could be 3D models, 2D cartoons, control points, or other control parameters. For simplicity the term "key shapes" will be used to refer to "mapped key visemes" in the remaining sections. Then, speech animations are synthesized by blending these key shapes using the optimal combinations of co-articulation models chosen by a dynamic programming technique.

Given a phoneme sequence PH1, PH2 PHn with timing labels, blending these key shapes generates intermediate animation frames. The simplest approach would be linear interpolation between adjacent pairs of key shapes ("linear co-articulation" and "linear interpolation" are interchangeable terms in this paper), but linear interpolation cannot provide natural co-articulation effects. Diphone and triphone co-articulation models are learned from the training data. How can these be combined to achieve optimal co-articulation effects? A dynamic programming technique is used to search for optimal combinations.

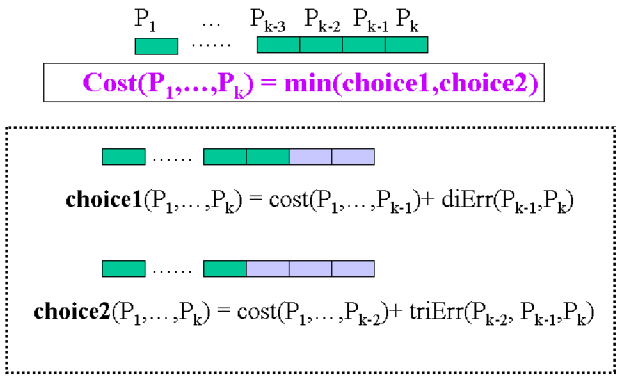

Figure 7: Illustration of novel speech synthesis using dynamic programming. Given an input phoneme sequence $P_1, P_2, \ldots, P_k$, its optimized cost is the minimum of thest two (assume the last two is a diphone co-articulation or the last three is a triphone co-articulation).

As mentioned in section 5, diphone and triphone co-articulation fitting errors are retained, and a constant linear co-articulation fitting error *LinearErr* is solved. So, the optimal sequence should minimize the total co-articulation errors. For example, given a phoneme sequence composed of PH_1, PH_2, and PH_3, there are two possible combinations: $diphone(PH_1 \rightarrow PH_2) + diphone(PH_2 \rightarrow PH_3)$ and $triphone(PH_1 \rightarrow PH_3)$. If the total co-articulation error of $triphone(PH_1 \rightarrow PH_3)$ is minimum, then the $triphone(PH_1 \rightarrow PH_3)$ co-articulation model is used to interpolate and provide minimum co-articulation error. Figure 7 illustrates this idea. This synthesis algorithm is very efficient, since its time complexity is linear $\Theta(N)$, where N is the number of phonemes.

7 Results and Evaluations

Three types of tests are designed to evaluate this approach. The first test is used to verify this approach by comparing ground-truth motion capture data and synthesized speech motion via trajectory comparisons. The second test is to compare this approach with some "common sense" and ad hoc interpolation methods, including linear and spline interpolation. In order to evaluate the applicability of this approach for 2D and keyframing animation, the third test is to generate human-like 2D cartoon speech animation, given key exaggerated cartoon shapes. Figure 11 shows some frames from the synthesized animation.

7.1 Ground-Truth Trajectory Comparisons

In the first evaluation, one arbitrary part of original speaker's audio (not used in previous training stage) is used to synthesize speech animation using this approach. The synthesized motion of 10 points around the mouth area are then compared frame-by-frame with the corresponding ground-truth data.

In this evaluation, manually picked key shapes are used that may not match the motion capture geometry perfectly. As such, the comparison of original and synthesized motions is only qualitative. Figure 8 shows the X, Y, and Z trajectory comparison results for a randomly picked point in the mouth region. As illustrated from figure 8, the learned co-articulations trajectories are similar to the real data.

7.2 Comparison with Ad Hoc Methods

This approach is also compared with a "common sense" approach (e.g. linear interpolation) and a de facto approach (e.g. spline inter-

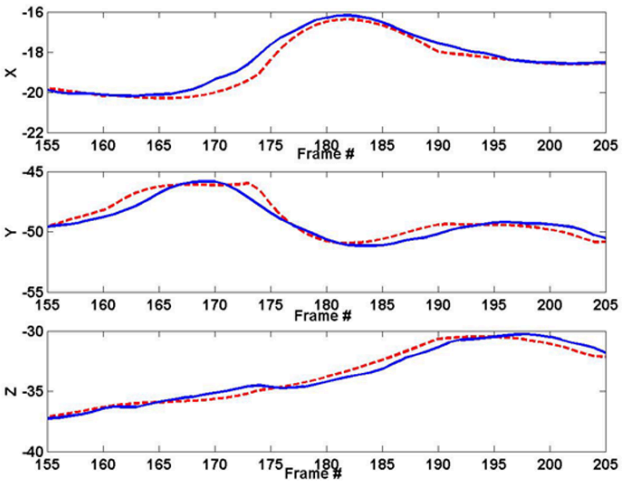

Figure 8: Trajectory comparisons with ground-truth motions. The solid line (blue) denotes ground-truth motion and the dashed line (red) denotes synthesized motion.

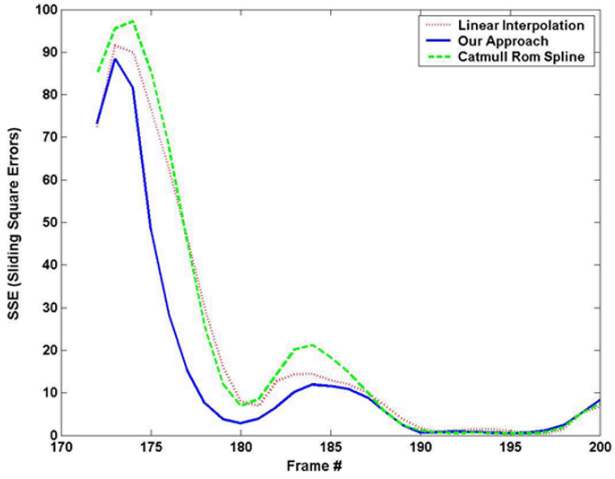

Figure 9: SSE curves corresponding to different approaches: the red dotted curve for linear interpolation, the green dashed curve for Catmull Rom spline, and blue solid curve for this new approach.

Figure 10: Illustration of designed keyshapes for 2D Cartoons.

polation) via "sliding" and total square errors. only the ten points around the mouth area are compared. To synthesize the motions of these ten points corresponding to test audio, some motion capture frames are manually picked as key shapes, then motions are synthesized with different approaches. *Sliding Square Errors* (SSE) and *Total Square Errors* (TSE) are defined as:

$$SSE_t = \sum_{i=1}^{10} (V_{t,i} - v_{t,i})^2 \tag{8}$$

$$TSE = \sum_{t=1}^{T} SSE_t \tag{9}$$

where $V_{t,i}$ is the position of i^{th} point of t^{th} frame in original motion capture data, $v_{t,i}$ is the position of i^{th} point of t^{th} frame in the synthesized motion, and T is the number of compared frames. Figure 9 illustrates SSE curves corresponding to different approaches (in Figure 9, TSE of linear interpolation is 609.73, TSE of Catmull Rom spline is 656.86, and TSE of this approach is 443.24).

7.3 7.3 2D Cartoon/Keyframing Applications

To demonstrate that this explicitly learned co-articulation model can be applied to keyframing applications, an animator is asked to draw some 2D key cartoons (Figure 10) and then used this approach to synthesize speech animation.

7.4 Eye Gaze, Eye Blink, Teeth, and Tongue

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 [14]is used to generate eye motion, and the multi-resolution hair modeling and rendering system [24] 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, several key tongue shapes are designed for phonemes and simple interpolation is used to generate the tongue motion.

8 Discussions and Conclusions

In this paper a novel approach is presented for learning speech co-articulation models from real motion capture data. By using a weight-decomposition method, this approach learns the personality (speech co-articulations) of a captured human subject. This personality can then be applied to any target faces. Instead of generating speech animation using ad hoc smoothing rules, the statistical models learned from real speakers make the synthesized speech animation more natural and human-like.

This approach can be used for various purposes. For animators, after initial key shapes are provided, this approach can serve as a fast tool to generate natural looking speech motion while simultaneously preserving the expressiveness of the animation. On top of the generated speech animation, animators can refine the animation by adjusting the visemes and timing as desired, to improve quality or produce stylized cartoon effects. When automated speech animation in real time speed is required, this approach can be used for this purpose. Because of the very compact size of the co-articulation models learned by this approach, it can be conveniently applied onto many mobile-game platforms, such as PDAs and cell phones.

The major limitation of this method is that it depends on "combinational sampling" from training data. The best results require that the training data have a complete set of phonemes and phoneme combinations. The learned co-articulation model probably depends on speech rates, since the co-articulation of fast speech may be quite different from that of normal speech.

This co-articulation modeling approach is efficient and reasonably effective for neutral speech, but it does not differentiate emotional states and speaking rates. As such, future work could be to extend the current co-articulation model to handle emotions and

different speaking rates, at the cost of requiring significantly more training data.

9 Acknowledgement

This research has been funded by the Integrated Media System Center/USC, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Special Thanks go to Tae-Yong Kim for providing hair modeling and rendering programs, Hiroki Itokazu and Bret St. Clair for model preparation, Pamela Fox for 2D cartoon design, and Murtaza Bulut for phoneme-alignment help. We also appreciate many valuable helps and comments from Jun-Yong Noh and Douglas Fidaleo and other colleagues in the CGIT Lab/USC.

References

[1] E. Agelfors, J. Beskow, B. Granstrom, M. Lundeberg, G. Salvi, K.E. Spens, and T. Ohman. Synthetic visual speech driven from auditory speech. *Proceedings of AVSP'99*, 1999.

[2] S. Basu, N. Oliver, and A. Pentland. 3d modeling and tracking of human lip motions. *ICCV'98*, pages 337–343, 1998.

[3] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum(Proceedings of Eurographics 2003)*, 22(3), 2003.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. *Proceedings of ACM SIGGRAPH'99*, pages 187–194, 1999.

[5] B. Bodenheimer, C. F. Rose, S. Rosenthal, and J. Pella. The process of motion capture: Dealing with the data. *Proceedings of Eurographics Workshop on Computer Animation and Simulation*, 1997.

[6] M. Brand. Voice pupperty. *Proceedings of ACM SIGGRAPH'99*, pages 21–28, 1999.

[7] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. *Proceedings of ACM SIGGRAPH'97*, pages 353–360, 1997.

[8] C. Bregler, L. Loeb, E. Chuang, and H. Despande. Turing to the masters: Motion capturing cartoons. *ACM Transaction on Graphics*, 21(3):399–407, 2002.

[9] B. W. Choe and H. S. Ko. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. *IEEE Computer Animation Conference*, pages 12–19, 2001.

[10] E. Chuang and C. Bregler. Performance driven facial animation using blendshape interpolation. *CS-TR-2002-02,Department of Computer Science, Stanford University*, 2002.

[11] M. M. Cohen and D. W. Massaro. Modeling coarticulation in synthetic visual speech. *Magnenat-Thalmann N., Thalmann D. (Editors), Models and Techniques in Computer Animation, Springer Verlag*, pages 139–156, 1993.

[12] D. Decarlo, D. Metaxas, and M. Stone. An antropometric face model using variational techniques. *Proceedings of ACM SIGGRAPH'98*, pages 67–74, 1998.

[13] Z. Deng, M. Bulut, U. Neumann, and S. S. Narayanan. Automatic dynamic expression synthesis for speech animation. In *Proc. of IEEE Computer Animation and Social Agents (CASA) 2004*, pages 267–274, Geneva, Switzerland, July 2004.

[14] Z. Deng, J. P. Lewis, and U. Neumann. Automated eye motion synthesis using texture synthesis. *IEEE Computer Graphics & Applications*, pages 24–30, March/April 2005.

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

[16] I. Essa, S. Basu, T. Darrell, and A. Pentland. Modeling, tracking and interactive animation of faces and heads using input from video. *IEEE Computer Animation Conference*, pages 68–79, 1996.

[17] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. *ACM Transaction on Graphics(Proceedings of ACM SIGGRAPH'02)*, pages 388–398, 2002.

[18] the University of Edinburgh FESTIVAL. http://www.cstr.ed.ac.uk/projects/festival/.

[19] B. L. Goff and C. Benoit. A text-to-audovisual-speech synthesizer for french. *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 2163–2166, 1996.

[20] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. *Proceedings of ACM SIGGRAPH'98*, pages 55–66, 1998.

[21] P. Y. Hong, Z. Wen, and T. S. Huang. Real-time speech-driven face animation with expressions using neural networks. *IEEE Trans. On Neural Networks*, 13(1):100–111, 2002.

[22] K. Kahler, J. Haber, and H. P. Seidel. Geometry-based muscle modeling for facial animation. In *Proc. of Graphics Interface'2001*, 2001.

[23] G. Kalberer and L. Van Gool. Face animation based on observed 3d speech dynamics. *IEEE Computer Animation Conference*, pages 20–27, 2001.

[24] T. Y. Kim and U. Neumann. Interactive multiresolution hair modeling and editing. *ACM Transaction on Graphics*, 21(3):620–629, 2002.

[25] S. Kshirsagar and N. M. Thalmann. Visyllable based speech animation. *Computer Graphics Forum (Proc. of Eurographics'03)*, 22(3), 2003.

[26] S. P. Lee, J. B. Badler, and N. Badler. Eyes alive. *ACM Transaction on Graphics*, 21(3):637–644, 2002.

[27] Y. C. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. *Proceedings of ACM SIGGRAPH'95*, pages 55–62, 1995.

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

[29] T. Masuko, T. Kobayashi, M. Tamuram, J. Masubuchi, and K. Tokuda. Text-to-speech synthesis based on parameter generation from hmm. *ICASSP'98*, pages 3745–3748, 1998.

[30] J. Y. Noh and U. Neumann. Expression cloning. *Proceedings of ACM SIGGRAPH'01*, pages 277–288, 2001.

[31] F. I. Parke and K. Waters. *Computer Facial Animation*. A K Peters, Wellesley, Massachusets, 1996.

[32] A. Pearce, B. Wyvill, G. Wyvill, and D. Hill. Speech and expression: A computer solution to face animation. *Proceedings of Graphics Interface'86*, pages 136–140, 1986.

[33] C. Pelachaud. Communication and coarticulation in facial animation. *Ph.D. Thesis, Univ. of Pennsylvania*, 1991.

[34] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. *Proceedings of ACM SIGGRAPH'98*, pages 75–84, 1998.

[35] S. Roweis. Em algorithm for pca and spca. *Neural Information Processing Systems (NIPS)'97*, pages 137–148, 1997.

[36] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis, and animation. *Journal of Visualization and Computer Animation*, 1(4):73–80, 1990.

[37] K. Waters and J. Frisble. A coordinated muscle model for speech animation. *Proceedings of Graphics Interface'95*, pages 163–170, 1995.

[38] J. J. Williams and A. K. Katsaggelos. An hmm-based speech-to-video synthesizer. *IEEE Trans. On Neural Networks*, 13(4):900–915, 2002.

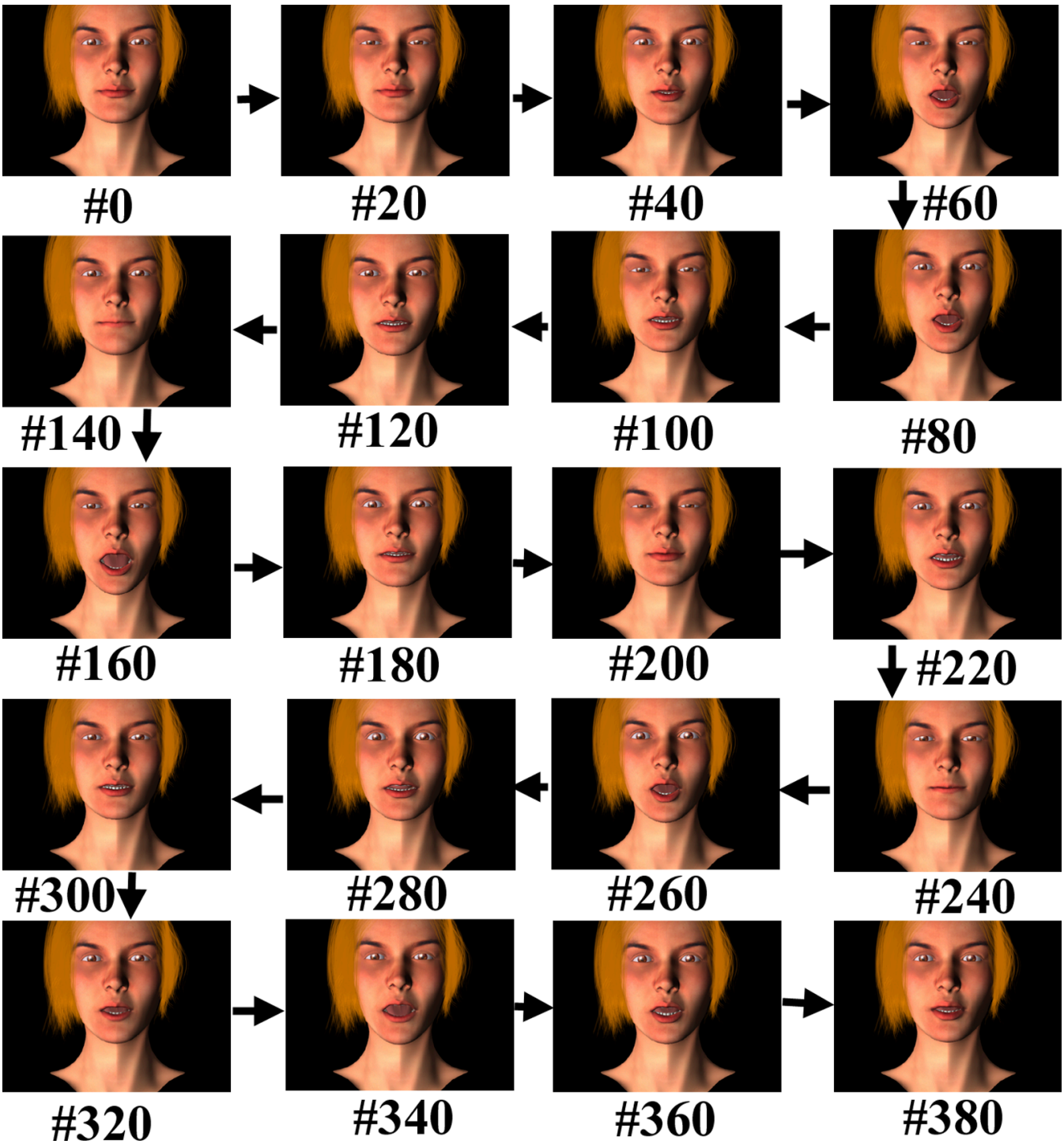

Figure 11: frames of synthesized talking (phrase is "With continued advances in facial animation and computer vision,natural-realistic communication and interaction between people via avatars integration is being realized").