Animating Lip-Sync Characters

Yu-Mei Chen[*] Fu-Chung Huang[†] Shuen-Huei Guan[*‡] Bing-Yu Chen[§]
Shu-Yang Lin[*] Yu-Hsin Lin[*] Tse-Hsien Wang[*]
[*§]National Taiwan University [†]University of California at Berkeley [‡]Digimax
[*]{yumeiohya,drake,young,b95705040,starshine}@cmlab.csie.ntu.edu.tw
[†]jonash@eecs.berkeley.edu [§]robin@ntu.edu.tw

ABSTRACT

Speech animation is traditionally considered as important but tedious work for most applications, especially when taking lip synchronization (lip-sync) into consideration, because the muscles on the face are complex and interact dynamically. Although there are several methods proposed to ease the burden on artists to create facial and speech animation, almost none are fast and efficient. In this paper, we introduce a framework for synthesizing lip-sync character speech animation from a given speech sequence and its corresponding text. Starting from training the **dominated animeme models** for each kind of phoneme by learning the animation control signals of the character through an EM-style optimization approach, and further decomposing the dominated animeme models to the polynomial-fitted animeme models and corresponding dominance functions while taking coarticulation into account. Finally, given a novel speech sequence and its corresponding text, a lip-sync character speech animation can be synthesized in a very short time with the dominated animeme models. The synthesized lip-sync animation can even preserve exaggerated characteristics of the character's facial geometry. Moreover, since our method can synthesize an acceptable and robust lip-sync animation in almost realtime, it can be used for many applications, such as lip-sync animation prototyping, multilingual animation reproduction, avatar speech, mass animation production, etc.

1. INTRODUCTION

With the popularity of 3D animation and video games, facial and speech animations are becoming more important than ever. Although many technologies have allowed artists to create high quality character animation, facial and speech animations are still difficult to sculpt, because the correlation and interaction of the muscles on the face are very complicated. Some physically-based simulation methods are provided to approximate the muscles on the face, but the computational cost is very high. A less flexible but affordable alternative is a performance-driven approach [40, 20, 26], where the motion of an actor is cross-mapped and transferred to a virtual character (see [32] for further discussion). This approach gains much success, but the captured performance is difficult to re-use and a new performance is required each time when creating a new animation or speech sequence. Manual adjustment is still a popular approach besides the above two, where artists are requested to adjust the face model controls frame by frame and compare the results back and forth.

When creating facial animation, lip synchronization (lip-sync) speech animation for a character model is more challenging, which requires much more labor and accuracy in timing for millisecond-precise key-framing. Given a spoken script, the artist has to first match the position of the lips at their supposed position. The *transitions* from word to word or phoneme to phoneme are even more important and need to be adjusted carefully. As opposed to simple articulated animation which can be key-framed with linear techniques, the transitions between lip shapes are non-linear and difficult to model.

The *transitions* from phoneme to phoneme, or *coarticulation*, play a major role in facial and speech animation [30, 15]. *Coarticulation* is the phenomenon where a phoneme can influence the mouth shape of the previous and next phonemes. In other words, the mouth shape depends not only just on the current phoneme itself but also on its context including at least the previous and next phonemes. Frequently this happens when a vowel influences a preceding or succeding consonant. Some previous methods have tried to model *coarticulation* using a strong mathematical framework, or to reduce it to a simpler version, but they are however complicated or insufficient to produce a faithful model.

In this paper, a framework is proposed to synthesize a lip-sync character speech animation from a given novel speech sequence and its corresponding text by generating animation control signals from the pre-trained **dominated animeme models**, which are obtained by learning the speech-to-animation control signals (e.g., the character controls used in Maya or similar modeling tools) with sub-phoneme accuracy for capturing *coarticulation* faithfully, and further decomposed to the polynomial-fitted animeme models and corresponding dominance functions according to the phonemes through an EM-style optimization approach. Rather than using absolute lip shapes for training as some previous work, the speech-to-animation control signals are used for better training/synthesis results and animation pipeline integration. Moreover, once there is no well-adjusted speech-to-

Computer Graphics Workshop 2010 Hsinchu, Taiwan
Copyright 2010 ACM X-XXXXX-XX-X/XX/XX ...$10.00.

animation control signal, we also provide a method to cross-map captured lip motion to the character, which can be the lip-tracking result from a speech video or a 3D lip motion captured by a motion capture device.

In the synthesis phase, given a novel speech sequence and its corresponding text, the **dominated animeme models** are composed to generate the speech-to-animation control signals automatically to synthesize a lip-sync character speech animation. This process only takes a very short time and can preserve the character's exaggerated characteristics. Since the synthesized speech-to-animation control signals can be used in Maya or similar modeling tools directly, our framework can be integrated into existing animation production pipelines easily. Moreover, since our method can synthesize an acceptable and robust lip-sync animation in almost realtime, it can be used in many applications for which prior techniques are too slow, such as lip-sync animation prototyping, multilingual animation reproduction, avatar speech, mass animation production, etc.

2. RELATED WORK

Face modeling and facial/speech animation generation are broad topics in computer graphics; [15, 30, 32] provide a good survey. In this section, we separate the face modeling and specific modeling for lips in the discussion.

2.1 Facial Animation and Modeling

Most facial animation and modeling methods can be categorized into parameterized/blend-shape, physically-based, data-driven, and machine-learning approaches. For parameterized/blend-shape modeling, faces are parameterized into controls; the synthesis is done manually or automatically via control adjustment. Previous work on linear blend-shape [17, 31, 4], face capturing/manipulation (FaceIK) [41], and face cloning/cross-mapping [29, 33, 6, 28, 36] provided a fundamental guideline for many extensions, however, the limitation of the underlying mathematical framework causes some problems, e.g., the faces outside the span of examples or parameters cannot be realistically synthesized, and the technique requires an excessive number of examples. There are also some methods for reducing the interference between the blend-shapes [24] or enhancing the capabilities of cross-mapping to animate the face models [13].

Physically-based methods simulate the muscles on the face, and the underlying interaction forms the subtle motion on the muscles. Previous methods [8, 34] have gained success in realism. The advantage of the physically-based methods over the parameterized/blend-shape ones is extensibility: the faces can be animated more realistically than other approaches, and the framework allows for interaction with objects. The muscle-simulation mathematical framework is, however, very complicated, and hence the cost for preparing and animating the 3D faces is higher.

Data-driven methods [14] form a database from a given very large training data set of faces. Subsequent faces are generated from searching the database with some constraints such as minimizing the discontinuity between the frames, and the path contained in the database forms a newly synthesized facial animation. The data-driven methods have to deal with missing training data or repetitive occurrence of the same records.

Machine-learning techniques base their capabilities on the learned statistical parameters from the training samples. Previous methods [1, 10, 39, 37] employed various mathematical models and can generate new faces from the learned statistics while respecting the given sparse observations of the new data.

2.2 Lip-Sync Speech Animation

Many speech animation methods derive from the facial animation and modeling techniques. The analysis of the **phonemes** under the context of speech-to-face correspondence, a.k.a. the **viseme**, is the subject of much successful work. Many previous methods addressed this issue with spline generation, path-finding, or signal concatenation.

Parameterized/blend-shape techniques [3, 2, 9] for speech animation are the most popular methods because of their simplicity. Sifakis *et al.* [35] presented a physical-based approach to simulate the speech controls based on their previous work [34] for muscle activation. This method can interact with objects while simulating, but still, the problem is the simulation cost. Data-driven approaches [5, 14] form a graph for searching the given sentences. Like similar data-driven approaches, they used various techniques, such as dynamic programming, to optimize the searching process. Nevertheless they still suffer from missing data or duplicate occurrence. Machine-learning methods [18, 7, 16, 22, 38] learn the statistics for phoneme-to-animation correspondence, which is called the **animeme**.

Löfqvist [25] and Cohen and Massaro [11] provided a key insight to decompose speech animation signal into target values and dominance functions to model *coarticulation.* The dominance functions are sometimes reduced to a diphone or triphone model [16] for simplicity. The original framework, however, shows examples such as a time-locked model or a look-ahead model that are difficult to explain by either the diphone or triphone model. Their methods are later extended by Cosi *et al.* [12] with shape functions and resistance functions, which are the basic concept for the **animeme**. Some recent methods [35, 22, 38] used the concept of animeme, a shape function, to model the sub-viseme signals to increase the accuracy of phoneme fitting.

Kim and Ko [22] extended [18] by modeling viseme within a smaller sub-phoneme range with a data-driven approach. However *coarticulation* is modeled via a smooth function in their regularization with parameters found empirically. Moreover, it has to resolve conflicting and insufficient records in the training set. Sifakis *et al.* [35] extended their previous work [34] to model the muscle control signal spline (the animeme, or they call it physemes) for each phoneme and concatenate these splines for words. Their result shows that each phoneme has various similar spline with slightly difference due to *coarticulation*, which is modeled using linear cross-fade weighting in a diphone or triphone fashion.

Wampler *et al.* [38] extended the multilinear face model [37] to derive new lip-shapes for a single face model. *Coarticulation* is modeled by minimizing the lips' position and forces exerted. However, it is usually unnecessary to sample the face tensor space to produce a single speech segment. Moreover, the face tensor space also inherits the curse of dimensionality, which is also a difficult topic for facial capture.

We learned from many successful advantages of previous methods and improved the deficiencies inherited from them. Cross-mapping eases the pain of 3D capture, and statistics are learned for constructing the animeme models rather than simply using them for performance-driven animation. The analysis in a sub-viseme, or so-called animeme, space has a significant improvement over the viseme analysis. Our method also decomposes the dominance function from the animeme model and extends coarticulation beyond a simple diphone or triphone model.

3. OVERVIEW

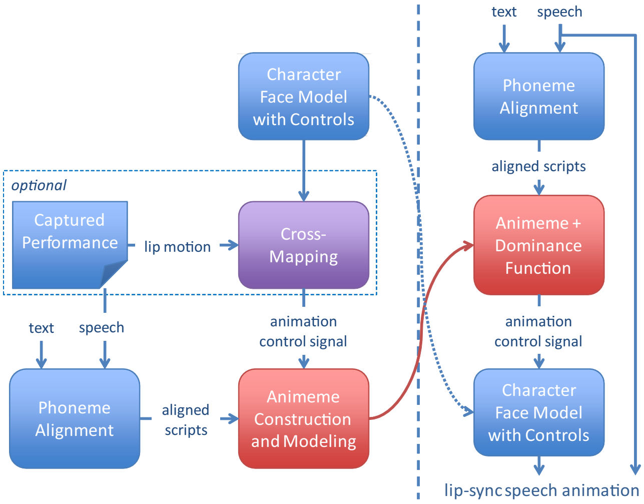

Figure 1: System flowchart.

Figure 1 shows our system flowchart. The system can be divided into two phases: training (left) and synthesis (right). In the training phase, the system takes a captured lip motion as the input, or directly uses the animation control signals made by animators. If we choose the lip-tracking result from a speech video or a 3D lip motion captured by a motion capture device, the data in the vertex domain will be cross-mapped to the control signal domain first in Section 4. Once there exists an acceptable lip-sync character animation, the previous capture and cross-mapping processes can be omitted, and the speech-to-animation control signal of the existed artist-sculpted or captured lip-sync character animation can be used directly.

Then, the speech captured with the lip motion and its corresponding text are aligned by using SPHINX-II [21] to obtain the aligned scripts, which contain the phonemes and their starting time stamps and durations in the speech. The aligned scripts and animation control signal C^i are used to construct the **dominated animeme models** in Section 5 that can be used for future reconstruction and synthesis.

In the synthesis phase, we take a novel speech and its corresponding text as the input and use SPHINX-II again to align the phonemes in the speech, which are used to query the animemes and dominance functions to compose the animation control signal C^* (Section 5). Finally, the animation control signal C^* is used to animate the character model in Maya or similar modeling tools to generate a lip-sync character speech animation.

4. CROSS-MAPPING

Although our system input is animation control signals, to ease the efforts for adjusting the character (lip) model, we also provide a method to cross-map the captured lip motion to the animation control signals. After the lip motion is captured, the key-lip-shapes L_k are identified first, which can be pointed out by the artist or by using an unsupervised clustering algorithm, affinity propagation [19]. The key-lip-shapes L_k are then used to fit the captured lip motion L^i for each frame i by using the Non-Negative Least Square (NNLS) algorithm [23] to obtain the blending coefficients α_k^i. This process can be expressed as the following constrained minimization:

$$\min \|L^i - \sum_{k=1}^{K} \alpha_k^i L_k\|^2, \quad \forall \alpha_k^i \geq 0,$$

where K is the number of identified key-lip-shapes. The above clustering and fitting process for the captured lip motion needs to be performed only once. If the target character model has some well-defined bases, it is better to assign the key-lip-shapes to the bases manually, since the blending coefficients α_k^i can be used as the control signals C^i directly without further processing.

To cross-map the input captured lip motion to the target character model (the Character Face Model with Controls in Figure 1), the identified key-lip-shapes L_k are first used to guide the artist to adjust the vertices $\mathbf{V}$ on the lips of the target character model to imitate the key-lip-shapes L_k while keeping the character's characteristics. The number of adjusted vertices should be equal to or more than that of character controls C (i.e., $\|\mathbf{V}\| \geq \|C\|$) for solving the constrained minimization in the next paragraph. Then, the blending coefficients α_k^i are used to blend the adjusted lip vertices $\mathbf{V}_k$ for key-lip-shapes L_k to obtain the lip vertices $\mathbf{V}^i$ for each frame i via

$$\mathbf{V}^i = \sum_{k=1}^{K} \alpha_k^i \mathbf{V}_k.$$

Instead of using the lip vertices $\mathbf{V}^i$ for training, for better training/synthesis results and animation pipeline integration, the training and synthesizing are performing on character controls. Hence, the NNLS algorithm is then used again to obtain the animation control signal C^i for each frame i by fitting the lip vertices $\mathbf{V}^i$ as the constrained minimization: $\min \|\mathbf{V}^i - \mathbf{V}_{C^i}\|^2$, where $\mathbf{V}_{C^i}$ denotes the same lip vertex set $\mathbf{V}$ deformed by the animation control signal C^i and each character control in C^i is constrained to 0∼1.

5. DOMINATED ANIMEME MODEL

To animate the character (face) model from a given script (phonemes) as shown in Figure 1, it is necessary to learn the relationship between the phonemes and the animation control signal C^i cross-mapped from the captured lip motion, which called animeme that means the animation representation of the phoneme. However, due to coarticulation, it is hard to model the animeme by a simple function, so we model the animation control signal C^i as a convolution of two functions: one is the function to fit the animeme, and the other is its dominance function.

Given an animation control signal C^i and its corresponding phoneme sequence (the aligned scripts in Figure 1), the signal can be treated as the summation of the animemes modulated by their dominance functions, which are corresponded with the given phoneme sequence. In mathematical formulation, the animation control signal C^i can be described as:

$$C^i = \sum_{j=1}^{J} D_j(i) A_j(i), \tag{1}$$

where $j = 1, 2, ..., J$ is the j-th phoneme in the given phoneme sequence, $A_j(t)$ and $D_j(t)$ are the function forms of the animeme and its dominance function of the j-th phoneme. Note that the phonemes farther away from the current phoneme may have very little contribution to it. In other words, the influence of modulating dominance functions far from it is relatively small. Hence, our goal is to construct and model the animeme $A_j(t)$ and its dominance function $D_j(t)$ for each phoneme j in the training phase. In the synthesizing phase, Eq. 1 can also be used to generate the animation control signal $C^* = C^i$ for each time step i by a given phoneme sequence, which can be used to animate the target character model in Maya or similar modeling tools.

5.1 Animeme Modeling

To solve Eq. 1 for simultaneously obtaining the animeme $A_j(t)$ and its dominance function $D_j(t)$ for the phoneme j is difficult. Hence, we first assume that the dominance function $D_j(t)$ is known and fixed as D_j^i and each phoneme appears in the phoneme sequence exactly only once.

The animeme $A_j(t)$ is modeled as a polynomial function, so the problem of modeling it is reduced to find the polynomial coefficients $a_j^0, a_j^1, ..., a_j^M$ for the animeme $A_j(t)$ as:

$$C^i = \sum_{j=1}^{J} D_j^i \left[\sum_{m=0}^{M} a_j^m (t_j^i)^m \right], \tag{2}$$

where s_j, d_j are the starting time stamp and the duration of j-th phoneme, and $t_j^i = (i - s_j)/d_j$ in order to normalize the duration of the phoneme. In our experiment, we use $M = 4$. Since we want to find the coefficients $a_j^0, a_j^1, ..., a_j^M$ for each phoneme j, in a regression manner, we can set the partial derivative of regression error $\mathbf{R}$ with respect to the m-th coefficient a_j^m from j-th phoneme to zero. The least square fitting for regression is:

$$\begin{aligned} f_i &= C^i - \sum_{j=1}^{J} D_j^i \left[\sum_{m=0}^{M} a_j^m (t_j^i)^m \right] \\ \mathbf{R} = \mathbf{F}^T\mathbf{F} &= \sum_{i=0}^{n} \left(C^i - \sum_{j=1}^{J} D_j^i \left[\sum_{m=0}^{M} a_j^m (t_j^i)^m \right] \right)^2, \end{aligned} \tag{3}$$

where $\mathbf{F}$ is the column-concatenated vector form for each element f_i. Since the unknowns a_j^m are linear in $\mathbf{F}$, the problem is essentially a linear least-square fitting.

By setting all partial derivatives to zero and arranging Eq. 3, we can obtain the following matrix representation:

$$\mathbf{D} = \begin{bmatrix} D_1^1 & D_1^1 t_1^1 & \cdots & D_1^1 (t_1^1)^M & \cdots & D_J^1 & \cdots & D_J^1 (t_J^1)^M \\ D_1^2 & D_1^2 t_1^2 & \cdots & D_1^2 (t_1^2)^M & \cdots & D_J^2 & \cdots & D_J^2 (t_J^2)^M \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ D_1^n & D_1^n t_1^n & \cdots & D_1^n (t_1^n)^M & \cdots & D_J^n & \cdots & D_J^n (t_J^n)^M \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a_1^0 & a_1^1 & \cdots & a_1^M & \cdots & a_J^0 & \cdots & a_J^M \end{bmatrix}^T$$

$$\mathbf{C} = \begin{bmatrix} C^0 & C^1 & C^2 & \cdots & C^n \end{bmatrix}^T,$$

where $\mathbf{D}$ is the dominance matrix, $\mathbf{A}$ is the coefficient vector we want to solve, and $\mathbf{C}$ is the observed values at each time i, so the minimum error to the regression fitting can be written in the standard normal equation with the following matrix form:

$$(\mathbf{D}^T\mathbf{D})\mathbf{A} = \mathbf{D}^T\mathbf{C}, \tag{4}$$

where $\mathbf{D}$ is an $n \times (M \times J)$ matrix, $\mathbf{C}$ is an n vector, and $\mathbf{A}$ is an $M \times J$ vector to be solved. Since $\mathbf{D}$ is a dense matrix, if we remove the assumption that each phoneme appears exactly once, J can be very large and makes the matrix $\mathbf{D}$ very huge. Furthermore, because multiple occurrence of a phoneme has to be fitted to the same value, we can arrange the multiple occurring terms and make it easier to solve. For example, if phoneme 1 appears twice as the first and third phonemes in the phoneme sequence, Eq. 2 becomes:

$$\begin{aligned} C^i &= D_{1_1}^i A_{1_1}(t_{1_1}^i) + D_2^i A_2(t_2^i) + D_{1_2}^i A_{1_2}(t_{1_2}^i) + ... \\ &= \left[D_{1_1}^i + D_{1_2}^i \right] a_1^0 + \left[D_{1_1}^i (t_{1_1}^i) + D_{1_2}^i (t_{1_2}^i) \right] a_1^1 + ... \\ &+ D_2^i a_2^0 + D_2^i a_2^1 (t_2^i) + D_2^i a_2^2 (t_2^i)^2 + ..., \end{aligned} \tag{5}$$

where 1_1 and 1_2 means the first and second times the phoneme 1 appeared. Note that the polynomial coefficients a_j^m of the animeme $A_j(t)$ are the same and independent to the occurrence. By the above re-arrangement, we can remove the original assumption that each phoneme can appear exactly only once, and rewrite the original entries in matrix $\mathbf{D}$ with the summation of each occurrence h of the same phoneme j as:

$$D_j^i (t_j^i)^m \Rightarrow \sum_h D_{j_h}^i (t_{j_h}^i)^m, \tag{6}$$

where j_h denotes the h-th time occurrence of the phoneme j.

5.2 Dominance Function

In the previous section, we were assuming that the dominance function $D_j(t)$ of j-th phoneme is known and fixed to estimate the animeme $A_j(t)$. In this section, we describe how to optimize the dominance function $D_j(t)$ over the regression, given that the animeme is known and fixed as A_j^i.

Some previous literatures [25, 16] described the dominance function as a bell-shape function. Extending [11], our dominance function $D_j(t)$ is basically modeled by an exponential function with a Gaussian form. That means, closer to the middle of the phoneme, the dominance function affects the lip shape for its own period, while it also simulates the influence for the previous and next phonemes if the frame moves toward to its tail. However, if we model the dominance function $D_j(t)$ with just a Gaussian form, it may also affect the previous and/or next phonemes strongly.

Hence, the dominance function $D_j(t)$ is modeled as follows:

$$D_j(t) = \begin{cases} \exp\left\{ -\left(\frac{t-\mu_j}{d_j \times \sigma_{1j} + \epsilon} \right)^2 \right\}, & |t - \mu_j| < \frac{d_j}{2}, \\ \exp\left\{ -\left(\frac{t-t_b}{d_j \times \sigma_{2j} + \epsilon} + \frac{t_b - \mu_j}{d_j \times \sigma_{1j} + \epsilon} \right)^2 \right\}, & otherwise, \end{cases} \tag{7}$$

where μ_j and d_j are the center and the duration of the phoneme j in a specific instance of occurrence that is given by the phoneme sequence, $t_b = \mu_j \pm d_j/2$ is the starting or ending time stamp of the phoneme j, ϵ is a small constant to prevent dividing by zero, and σ_{1j} and σ_{2j} are the influence controls which are the unknowns we want to solve. The first Gaussian form with σ_{1j} stands for the level of keeping the current phoneme's own shape, and the other Gaussian form with σ_{2j} represents the level to affect the neighboring phonemes.

Here, we want to solve the regression (Eq. 3) again as we did in the previous section. However, since the parameters σ_{1j} and σ_{2j} for regression are not linear, it requires more sophisticated solver and standard Gauss-Newton iterative solver [27] is used to approach the minimum of regression error $\mathbf{R}$. As we defined the residual error in the previous section, the Gauss-Newton algorithm linearizes the residual error as:

$$\begin{aligned} f_i &= C^i - \sum_{j=1}^{J} D_j(t^i)A_j^i \\ \mathbf{F}(\sigma_j + \delta) &\approx \mathbf{F}(\sigma_j) + \mathbf{J}\delta, \end{aligned} \tag{8}$$

where $t^i = i$, $\mathbf{F}$ is formed by f_i but takes the influence control $\sigma_j \in \{\sigma_{1j}, \sigma_{2j}\}$ for j-th phoneme as the input, δ is the updating step for gradient direction of the Gauss-Newton solver, and $\mathbf{J}$ is the Jacobian matrix. Each iteration of Gauss-Newton algorithm solves a linearized problem to Eq. 3, and after removing terms that are not dependent on δ, we get the following:

$$\begin{aligned} \mathbf{J}^T\,\mathbf{J}\,\delta &= -\mathbf{J}^T\,\mathbf{F} \\ \sigma_j^{k+1} &= \sigma_j^k + \delta. \end{aligned} \tag{9}$$

The Gauss-Newton algorithm repeatedly optimizes the regression error by updating δ to $\sigma_j^k \in \{\sigma_{1j}^k, \sigma_{2j}^k\}$ at the k-th iteration, and achieves linear convergence.

5.3 Animeme Construction

In the previous two sections, the estimation of the animeme $A_j(t)$ and the optimization of the dominance function $D_j(t)$ are described over the regression. Since the entire formulation is not linear and cannot be solved intuitively, we employed an EM-style strategy that iterates between the estimation of the animeme $A_j(t)$ and the optimization of the dominance function $D_j(t)$.

- The **E-step** involves estimating the polynomial coefficients a_j^m for each animeme $A_j(t)$ by solving a linear regression using standard normal equation.

- The **M-step** tries minimizing the regression error to estimate the influence controls σ_{1j} and σ_{2j} by improving the non-linear dominance function $D_j(t)$.

When the first time solving for **E-step**, the initial influence control parameters σ_{1j} and σ_{2j} involved in $D_j(t)$ are set to 1. At the **M-step**, where the Gauss-Newton algorithm linearizes the function with iteratively updating the influence controls σ_{1j} and σ_{2j}, all parameters of the polynomial coefficients a_j^m are carried from the first half of the iteration. The EM-style strategy keeps iterating between **E-step** and **M-step** until no more improvement on regression error can be done. Convergence of optimizing $D_j(t)$ is fast, but the effect of estimating $A_j(t)$ has more perturbation on σ_{1j} and σ_{2j}. Generally convergence involves hundreds of iterations, the process is, however, off-line computation in the training phase.

Table 1: The models used in this paper and the accompanying video.

model	vertex#	face#	control#
fat woman	5,234	5,075	7
boy	6,775	6,736	7
old hero	8,883	8,738	8
court lady	1,306	1,307	7

6. RESULT

Figure 2 shows a comparison of the signal fitted by our dominated animeme model (DAM), Cohen-Massaro model [11], and the multi-dimensional morphable model (MMM) [18] with the captured one. Note that the Cohen-Massaro model is implemented using our dominated animeme model by setting $M = 0$ in Eq. 3, i.e., the polynomial form is changed to only the constant term. The formulation of our dominance function (Eq. 7) is very similar to the authors' original form but with the flexible extension that the shapes of the phonemes can be varied. The reconstruction result of the Cohen-Massaro model is too smooth at some peaks of the captured data, such that consecutive phonemes are greatly influenced, i.e., they span too much. The fitted signal exhibits low frequency behavior, but the high frequency features are not as prominent as they should be. In contrast, our dominated animeme model spans more properly in range with respect to the training data. The multi-dimensional morphable model formulates the fitting problem and synthesis as a regulation problem. They fit each phoneme as a multidimensional Gaussian distribution and form the words or sentences as a path going through these phoneme regions by minimizing an energy function containing a target term and a smoothness term. The speech poses using multi-dimensional morphable model have good timing but lack prominent features, while our results reach closer to the peaks of the training data.

The average L^2 error norms for our dominated animeme model (DAM), Cohen-Massaro model, and multi-dimensional morphable model (MMM) are 0.406, 1.427, and 0.860, respectively. The Cohen-Massaro model produces 251% more error than our dominated animeme model, and the multi-dimensional morphable model, though slightly better, still produces 112% more error. The two kinds of error sources are: (a) inaccurate and/or imprecise timing; (b) signal strength is not high enough to be representative. Both errors are important in modeling coarticulation, but the first one is more severe, since it can interfere with understanding of the context.

The captured lip motion in the training phase involves 40 sentences, and about 5 minutes of speech context with unbiased content. In most cases, each phoneme occupies about 9∼12 ms., so our training base is sufficient to cover the fitting. In the training phase, constructing the dominated animeme model costs about 50∼60 minutes per control on a desktop PC with an Intel Core2 Quad Q9400 2.66GHz CPU and 4GB memory. For synthesizing a lip-sync speech animation, the animation control signal formed by our dominated animeme models is generated in realtime. Table 1 shows the number of vertex, face, and control of each model used in this paper and the accompanying video, respectively.

Figure 3shows a result of speaking a word - ”homework”. Different from other performance-driven or data-driven ap-

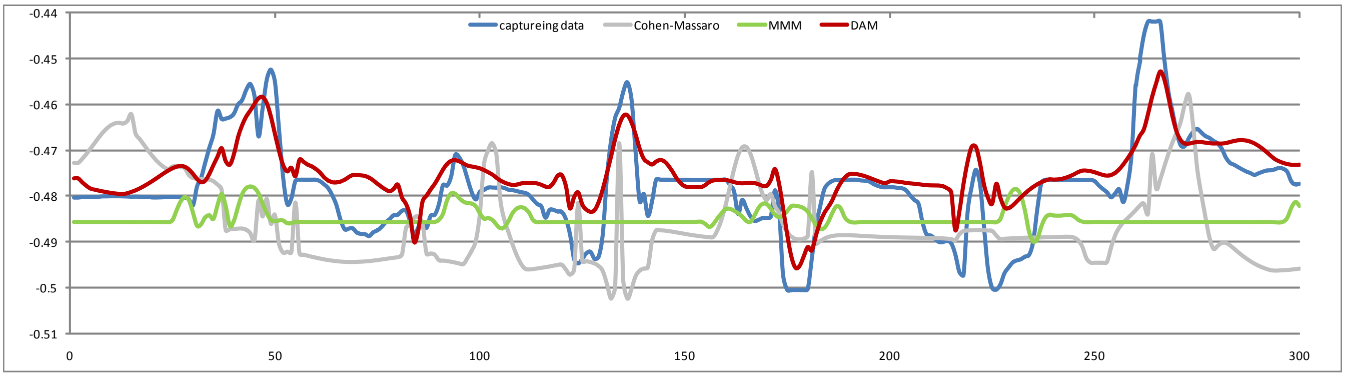

Figure 2: This graph shows a comparison of the signal fitted by Dominated Animeme Model (DAM), Cohen-Massaro Model, and Multi-dimensional Morphable Model (MMM) with the captured one. The value of y-axis is one of the coordinate of a feature around lip.

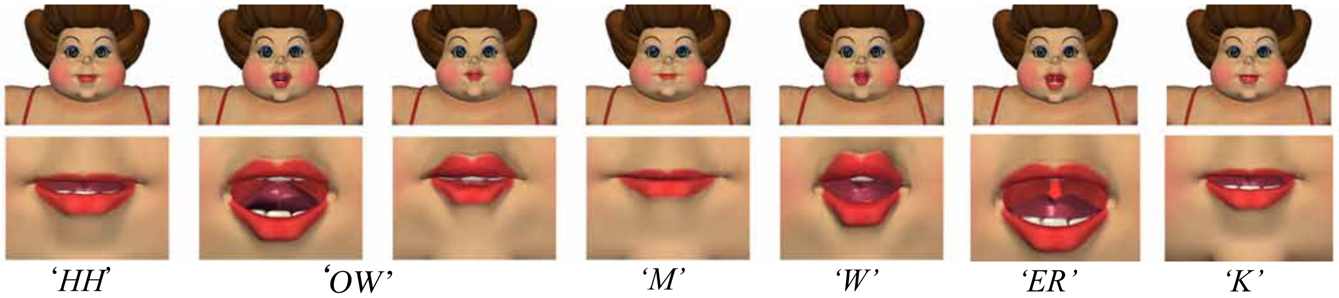

Figure 3: The result of speaking a word - ”homework” and its corresponding phonemes.

proaches, our method is actor and character independent, so we can use several kinds of characters. Since our dominated animeme model (DAM) and the multi-dimensional morphable model (MMM) have better signal fitting results shown in Figure 2, a visual comparison of the two models is performed as shown in Figure 4 by speaking a word - ”infringement” using the fat woman model. The close-up view of the mouth and its corresponding phonemes are also shown in the figure. By comparing the close-up view of the mouth, our dominated animeme model can perform better result than multi-dimensional morphable model, especially for $'F'$. By extending the phoneme dictionary, our method can also be used to produce multilingual lip-sync speech animations. Readers should refer to the accompanying video to see motion dynamics.

7. CONCLUSION AND FUTURE WORK

In this paper, we propose a new framework for synthesizing a lip-sync character speech animation with a given novel speech sequence and its corresponding text. Our method produces fairly nice transitions in time and generates the animation control parameters that are formed by our **dominated animeme model**, which is constructed and modeled from the training data in sub-phoneme accuracy for capturing *coarticulation* well. Through an EM-style optimization approach, the dominated animeme model is decomposed to the polynomial-fitted animeme models and corresponding dominance functions according to the phonemes. Given a phoneme sequence, the dominated animeme model is used to generate the animation control signal to animate the character model in Maya or similar modeling tools in a very short time while still keeping the character's exaggerated characteristics. Moreover, the **dominated animeme model** is constructed by the character controls instead of the absolute lip shapes, so it can perform better training/synthesizing result and is suitable to be integrated into the existed animation pipeline.

Even though the quality of the synthesized lip-sync character speech animation may not be perfect as compared with that of animation created manually by an artist, the synthesized animation can still easily be fine-tuned, since the automatically generated animation control signal is lip-synchronized and can be used directly in Maya or similar animation tools. Hence, our framework can be integrated into existed animation production pipelines easily. By extending the phoneme dictionary, our method can also be used to produce multilingual lip-sync speech animations easily. Furthermore, since our method can synthesize an acceptable and robust lip-sync character animation in almost realtime, it can be used for many applications for which prior methods are inadequate, such as lip-sync animation prototyping, multilingual animation reproduction, avatar speech, mass animation production, etc.

Our model still has some weaknesses, such as that it currently infers the dynamics of motion solely from the training data set. If the training data set does not contain speech similar to the synthesis target, results may be inaccurate. For example, if the training set contains only ordinary speech, it will be unsuitable for synthesizing a singing character, because the typical phoneme behavior for song varies

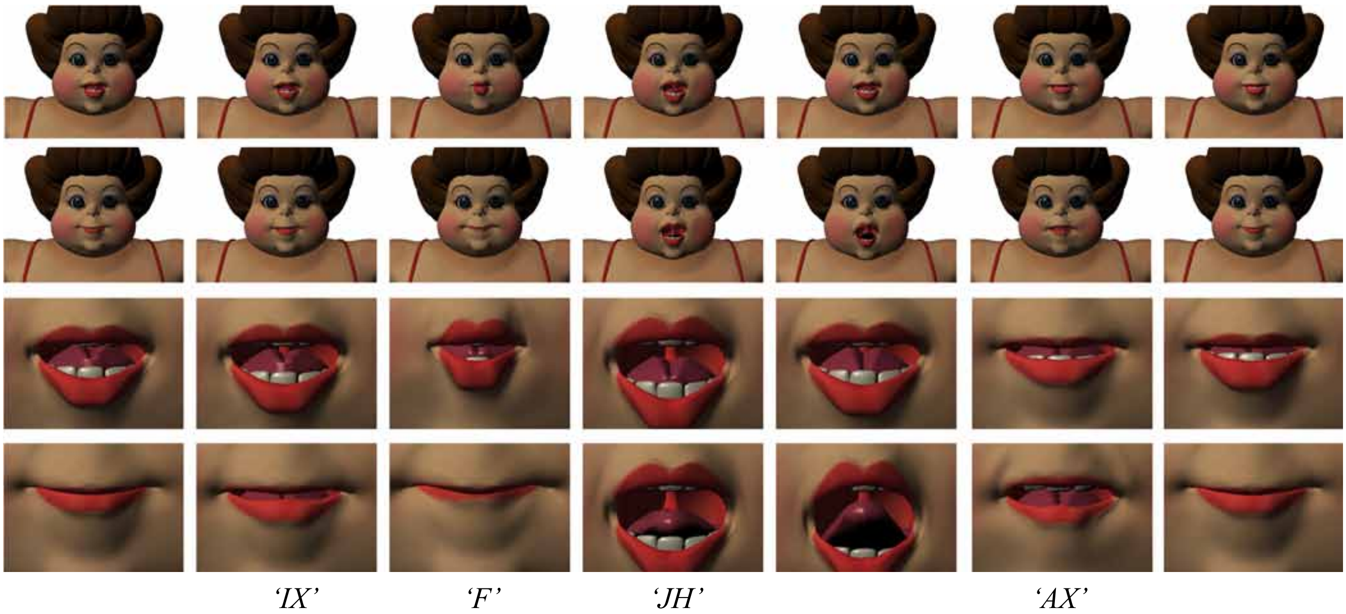

Figure 4: A visual comparison of dominated animeme model (upper) and multi-dimensional morphable model (lower) by speaking a word - "infringement".

greatly from ordinary speech and imposes more challenges for dynamics modeling.

A second weakness is that in our dominated animeme model, we used a function of Gaussian form to model the dominance function. The potential problem is that in song, certain phonemes may extend indefinitely with dragging sounds. It is not only difficult for a speech recognizer to identify the ending time, but also the Gaussian form cannot accommodate such effects. One possible solution is to model the dominance function with greater variability and non-symmetric models.

8. REFERENCES

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *ACM SIGGRAPH 1999 Conference Proceedings*, pages 187–194, 1999.

[2] M. Brand. Voice puppetry. In *ACM SIGGRAPH 1999 Conference Proceedings*, pages 21–28, 1999.

[3] C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. In *ACM SIGGRAPH 1997 Conference Proceedings*, pages 353–360, 1997.

[4] I. Buck, A. Finkelstein, C. Jacobs, A. Klein, D. H. Salesin, J. Seims, R. Szeliski, and K. Toyama. Performance-driven hand-drawn animation. In *Proceedings of the 2002 International Symposium on Non-Photorealistic Animation and Rendering*, pages 101–108, 2000.

[5] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin. Real-time speech motion synthesis from recorded motions. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 345–353, 2004.

[6] J. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 193–206, 2003.

[7] Y.-J. Chang and T. Ezzat. Transferable videorealistic speech animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 143–151, 2005.

[8] B. Choe, H. Lee, and H.-S. Ko. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation*, (2):67–79, 2001.

[9] E. Chuang and C. Bregler. Mood swings: expressive speech animation. *ACM Transactions on Graphics*, 24(2):331–347, 2005.

[10] E. S. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Pacific Graphics 2002 Conference Proceedings*, pages 68–76, 2002.

[11] M. M. Cohen and D. W. Massaro. Modeling coarticulation in synthetic visual speech. In *Computer Animation 1993 Conference Proceedings*, pages 139–156, 1993.

[12] P. Cosi, E. M. Caldognetto, G. Perin, and C. Zmarich. Labial coarticulation modeling for realistic facial animation. In *Proceedings of the 2002 IEEE International Conference on Multimodal Interfaces*, pages 505–510, 2002.

[13] Z. Deng, P.-Y. Chiang, P. Fox, and U. Neumann. Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, pages 43–48, 2006.

[14] Z. Deng and U. Neumann. efase: Expressive facial animation synthesis and editing with phoneme-isomap control. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 251–259, 2006.

[15] Z. Deng and U. Neumann. *Data-Driven 3D Facial Animation.* Springer, 2008.

[16] Z. Deng, U. Neumann, J. Lewis, T.-Y. Kim, M. Bulut,

and S. Narayanan. Expressive facial animation synthesis by learning speech coarticulation and expression spaces. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1523–1534, 2006.

[17] P. Ekman and W. V. Friesen. *Manual for the Facial Action Coding System.* Consulting Psychologist Press, 1977.

[18] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. *ACM Transactions on Graphics*, 21(3):388–398, 2002. (SIGGRAPH 2002 Conference Proceedings).

[19] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

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

[21] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The sphinx-ii speech recognition system: An overview. *Computer Speech and Language*, 7(2):137–148, 1993.

[22] I.-J. Kim and H.-S. Ko. 3d lip-synch generation with data-faithful machine learning. *Computer Graphics Forum*, 26(3):295–301, 2007. (Eurographics 2007 Conference Proceedings).

[23] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice-Hall, 1974.

[24] J. P. Lewis, J. Mooser, Z. Deng, and U. Neumann. Reducing blendshape interference by selected motion attenuation. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, pages 25–29, 2005.

[25] A. Löfqvist. *Speech Production and Speech Modeling*, chapter Speech as audible gestures, pages 289–322. Kluwer Academic Print on Demand, 1990.

[26] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics*, 27(5):1–10, 2008. (SIGGRAPH Asia 2008 Conference Proceedings).

[27] K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. Technical report, Technical University of Denmark, 2004.

[28] K. Na and M. Jung. Hierarchical retargetting of fine facial motions. *Computer Graphics Forum*, 23(3):687–695, 2004. (Eurographics 2004 Conference Proceedings).

[29] J.-Y. Noh and U. Neumann. Expression cloning. In *ACM SIGGRAPH 2001 Conference Proceedings*, pages 277–288, 2001.

[30] F. I. Parke and K. Waters. *Computer Facial Animation, 2nd Ed.* AK Peters, 2008.

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

[32] F. Pighin and J. P. Lewis. Performance-driven facial animation: Introduction. In *ACM SIGGRAPH 2006 Conference Course Notes*, 2006.

[33] H. Pyun, Y. Kim, W. Chae, H. W. Kang, and S. Y. Shin. An example-based approach for facial expression cloning. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 167–176, 2003.

[34] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics*, 24(3):417–425, 2005. (SIGGRAPH 2005 Conference Proceedings).

[35] E. Sifakis, A. Selle, A. Robinson-Mosher, and R. Fedkiw. Simulating speech with a physics-based facial muscle model. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 261–270, 2006.

[36] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405, 2004. (SIGGRAPH 2004 Conference Proceedings).

[37] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433, 2005. (SIGGRAPH 2005 Conference Proceedings).

[38] K. Wampler, D. Sasaki, L. Zhang, and Z. Popović. Dynamic, expressive speech animation from a single mesh. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 53–62, 2007.

[39] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. *Computer Graphics Forum*, 23(3):677–686, 2004. (Eurographics 2004 Conference Proceedings).

[40] L. Williams. Performance-driven facial animation. In *ACM SIGGRAPH 1990 Conference Proceedings*, pages 235–242, 1990.

[41] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558, 2004. (SIGGRAPH 2004 Conference Proceedings).