Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

eFASE: Expressive Facial Animation Synthesis and Editing with Phoneme-Isomap Controls

Zhigang Deng[1] and Ulrich Neumann[2]

[1]University of Houston
[2]University of Southern California

Abstract

This paper presents a novel data-driven system for expressive facial animation synthesis and editing. Given novel phoneme-aligned speech input and its emotion modifiers (specifications), this system automatically generates expressive facial animation by concatenating captured motion data while animators establish constraints and goals. A constrained dynamic programming algorithm is used to search for best-matched captured motion nodes by minimizing a cost function. Users optionally specify "hard constraints" (motion-node constraints for expressing phoneme utterances) and "soft constraints" (emotion modifiers) to guide the search process. Users can also edit the processed facial motion node database by inserting and deleting motion nodes via a novel phoneme-Isomap interface. Novel facial animation synthesis experiments and objective trajectory comparisons between synthesized facial motion and captured motion demonstrate that this system is effective for producing realistic expressive facial animations.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.6.8 [Simulation and Modeling]: Types of Simulation;

1. Introduction

In the entertainment industry the creation of compelling facial animation is a painstaking and tedious task, even for skilled animators. Animators often manually sculpt keyframe faces every two or three frames. Facial motion capture is widely used to acquire human facial motion data, but it remains difficult to modify and edit captured facial motion data to achieve animation goals or synthesize novel facial animations.

In this paper, we developed a novel data-driven expressive Facial Animation Synthesis and Editing system (eFASE) that generates expressive facial animations by concatenating captured facial motion data while animators establish constraints and goals. Its algorithm synthesizes an expressive facial motion sequence by searching for best-matched motion capture frames in the database, based on the new speech phoneme sequence, user-specified constrained expressions for phonemes and emotion modifiers.

Users can browse and select constrained expressions for phonemes using a novel 2D expressive phoneme-Isomap visualization and editing interface. Users can also optionally specify emotion modifiers over arbitrary time intervals. These user interactions are phoneme aligned to provide intuitive speech-related control. It should be noted that user input is not needed to create motion sequences, only to impart them with a desired expressiveness. Figure 1 illustrates the high-level components of the eFASE system.

Besides the effective search algorithm and intuitive user controls, our system provides novel and powerful editing tools for managing a large facial motion capture database. Since facial motion capture is not perfect, contaminated marker motions can occasionally occur somewhere in a motion capture sequence. Eliminating these contaminated motions is difficult but very useful. Our phoneme-Isomap based editing tool visualizes the facial motion database in an intuitive way, which can help users to remove contaminated motion sequences, insert new motion sequences intuitively, and reuse captured uncontaminated motions efficiently.

The contributions of this work include: (1) The introduced

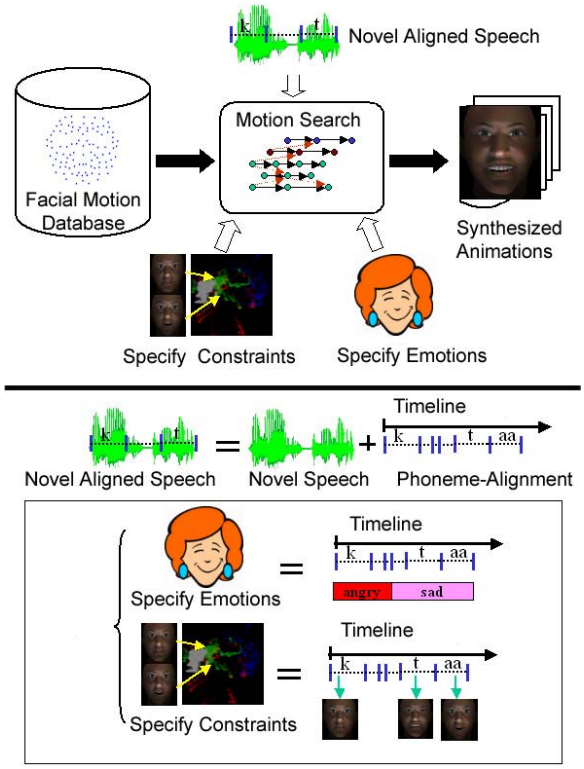

Figure 1: *Illustration of the eFASE pipeline. At the top, given novel phoneme-aligned speech and specified constraints, this system searches for best-matched motion nodes in the facial motion database and synthesizes expressive facial animation. The bottom illustrates how users specify motion-node constraints and emotions with respect to the speech timeline.*

motion-node constraints and emotion modifiers provide intuitive controls for data-driven facial animation synthesis and editing; (2) The novel phoneme-Isomap based control provides powerful editing tools for managing a large facial motion database.

The remainder of this paper is organized as follows: Section 2 reviews previous and related work on motion capture and facial animation. Section 3 describes the capture and processing of expressive facial motion data. Section 4 describes the construction of 2D expressive phoneme-Isomaps that allow users to interactively specify phoneme expression constraints and edit the motion database. Section 5 details how to perform motion editing operations and specify constraints for facial animation synthesis. Section 6 describes how to search for best-matched motion frames from the processed motion database to create complete animations while satisfying user-specified constraints. Finally, results (Section 7) and conclusions (Section 8) are presented.

2. Previous and Related Work

Various facial modeling and animation techniques have been proposed [PW96]. Physics-based methods [WF95, LTW95, KHS01] drive mouth movement by simulating the facial muscles. Performance-driven facial animation techniques [Wil90, CXH03] track facial motion of real performers and drive 3D face models accordingly. [GGW*98, PHL*98, ZSCS04] use blendshapes or traverse faces modeled from photographs or video streams to generate facial animations. Other approaches were presented to transfer existing animations to other face models [NN01, PKC*03, VBPP05, SNF05, DCFN06] and learn morphable face models to animate faces in images and video [BV99, BBPV03].

Data-driven facial animation approaches concatenate phoneme or syllable segments [BCS97, CG00, CFKP04, KT03, MCP*05] or modeling speech co-articulation from data [Lew91, Pel91, CM93, DLN05, KP05, DNL*06]. For example, recorded triphone video segments [BCS97] or syllable motion segments [KT03] are smoothly concatenated to synthesize novel speech motion. Rather than restricting within triphones or syllables, longer (≥ 3) phoneme sequences are combined in an optimal way using various search methods including greedy search [CFKP04] or the Viterbi search algorithm [CG00, MCP*05]. Different from the above pre-recorded motion recombinations, [Bra99, EGP02, CDB02, CB05] learn statistical models from real data for facial animation synthesis and editing. These above approaches can achieve synthesis realism, but their versatility and control are limited. One of their common limitations is that it is difficult to have expression control and intuitive editing without considerable efforts.

Our eFASE system employs a constrained dynamic programming algorithm, similar to [CG00, MCP*05], to search for the best-matched motion capture frames in the database. But the distinctions of our search algorithm include: (1) It introduces a new position velocity cost for favoring smooth paths. (2) By introducing an emotion mismatch penalty, our algorithm can seamlessly synthesize expressive facial animation, instead needing to create separate facial motion database for each emotion category, as previous approaches have done. (3) It introduces motion-node constraints and emotion modifiers into the search process, which make the control of data-driven facial animation synthesis feasible and intuitive.

3. Data Capture and Processing

A VICON motion capture system was used to capture expressive facial motion at a 120 Hz sample rate. An actress with 102 markers on her face was directed to speak a custom designed corpus composed of 225 phoneme-balanced sentences four times. Each repetition was spoken with a different expression (neutral, happy, angry and sad). Simulta-

neous facial motion and audio were recorded. Note that sentences for each emotion repetition are slightly different, because the actress could not speak some sentences with all four emotions. The total data include more than 105,000 motion capture frames (approximately 135 minutes recorded time). Due to occlusions caused by tracking errors (rapid large head movement accompanying expressive speech can cause markers to be tracked incorrectly) and the removal of unnecessary markers, we kept 90 of 102 markers for this work. (The 90 markers were fully tracked.) Figure 2 shows the 102 captured markers and the 90 kept markers. The motion frames for each corpus repetition are labeled with the intended expression, the only tag information required by the algorithm. Except for 36 sentences that are used for cross-validation and test comparisons, the other captured facial motion data are used for constructing the training facial motion database.

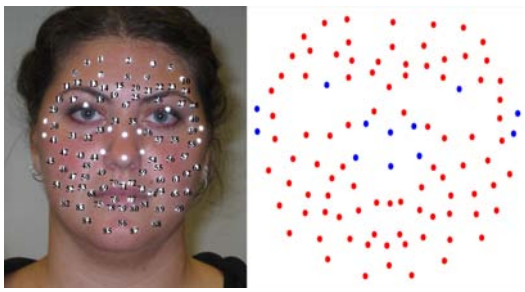

Figure 2: *The left is a snapshot of the captured actress. Blue and red points in the right panel represent the 102 captured markers, where the red points are the 90 markers used for this work.*

After data capture, we normalized the facial motion data. All the markers were translated so that a specific marker was at the local coordinate center of each frame. Then a statistical shape analysis method [BDNN05] was used to calculate head motion. The *Festival* speech recognition system [fes04] was used to perform automatic phoneme alignment on the captured audio. Accurate phoneme-alignment is important to the success of this work, and automatic phoneme-alignment is imperfect, so two linguistic experts manually checked and corrected all the phoneme-alignments by examining the corresponding spectrograms.

After head motion was removed from the motion capture data and the motions of all 90 markers in one frame were packed into a 270 dimensional motion vector, Principal Component Analysis (PCA) is applied onto all the motion vectors to reduce its dimensionality. We experimentally set the reduced dimensionality to 25, which covers 98.53% of the variation. Therefore, we transformed each 270-dimensional motion vector into a reduced 25-dimensional vector concatenating the retained PCA coefficients. In this paper, we use *Motion Frames* to refer to these PCA coefficient vectors or their corresponding facial marker configurations.

To make the terms used in this paper consistent, we defined two new terms: *Motion Nodes* and *Phoneme Clusters*. Based on the phonemes' time boundaries (from the above phoneme-alignment), we chopped the motion capture sequences into small subsequences that span several to tens of motion frames, and each subsequence corresponds to the duration of a specific phoneme. Each phoneme occurs many times in the spoken corpus, with varied co-articulation. We refer to these subsequences as *Motion Nodes*. For each motion node, its triphone context that includes its previous phoneme and next phoneme is also retained. Putting all motion nodes of a specific phoneme together produces thousands of motion frames representing the facial configurations that occur for this phoneme. All the motion-frames corresponding to a specific phoneme are referred to as a *Phoneme Cluster*. Each motion-frame in a phoneme cluster has an emotion label and a relative time property (relative to the duration of the motion node that it belongs to). The specific phoneme that a motion node represents is called *the phoneme of this motion node*. Fig. 3 illustrates the process of constructing phoneme clusters and motion nodes.

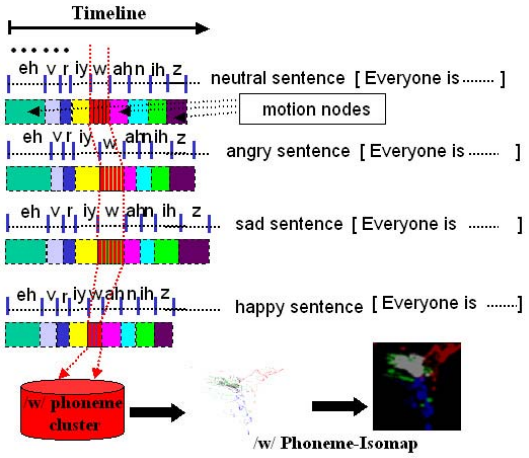

Figure 3: *To construct a specific /w/ phoneme cluster, all expressive motion capture frames corresponding to /w/ phonemes are collected, and the Isomap embedding generates a 2D expressive Phoneme-Isomap. Colored blocks in the figure are motion nodes.*

Besides the above phoneme clusters, we also built a facial motion-node database. The processed motion node database can be conceptually regarded as a 3D space (spanned by *sentence*, *emotion*, and *motion node order*). Because the sentence is the atomic captured unit, each motion node o_i (except the first/last motion node of a sentence recording) has a predecessor motion node $pre(o_i)$ and a successive motion node $suc(o_i)$ in its sentence (illustrated as solid direc-

tional lines in Fig. 4). Possible transitions from one motion node to another motion node are illustrated as dashed directional lines in Fig. 4. Note that motion nodes for the silence phoneme /pau/ were discarded, and if the /pau/ phoneme appears in the middle of a sentence's phoneme transcript, it will break the sentence into two sub-sentences when constructing the motion node database. Figure 4 illustrates the organization of the processed motion node database.

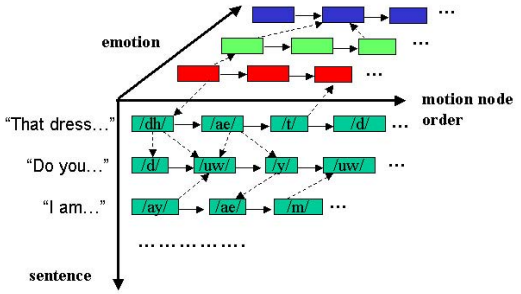

Figure 4: *Schematic illustration of the organization of the processed motion node database. Here solid directional lines indicate predecessor/successor relations between motion nodes, and dashed directional lines indicate possible transitions from one motion node to the other. The colors of motion nodes represent different emotion categories of the motion nodes.*

4. Expressive Phoneme Isomaps

This section describes how the phoneme clusters are transformed into 2D expressive phoneme-Isomaps. The phoneme-Isomaps are needed to allow users to interactively browse and select motion-frames. Similar to the application of PCA to a specific type of human body motion (e.g. *jumping*) to generate a low-dimensional manifold [SHP04], each phoneme cluster is processed with the Isomap framework [TSL00] to embed the cluster in a two-dimensional manifold (the neighbor number is set to 12).

We compared 2D Phoneme-PCA maps (two largest eigenvector expanded spaces) with 2D Phoneme-Isomaps. By visualizing both in color schemes, we found that points for one specific color (emotion) were distributed throughout the 2D PCA maps, and thus, the 2D PCA display is not very useful as a mean for frame selection. The 2D Phoneme-Isomaps cluster many of the color (emotion) points leading to a better projection, so that the points from the various expressions are better distributed and make more sense. We also found that directions, such as a vertical axis, often corresponded to intuitive perceptual variations of facial configurations, such as an increasingly open mouth. Figure 5 compares PCA projection and Isomap projection on the same phoneme clusters.

The above point-rendering (Fig. 5) of 2D expressive

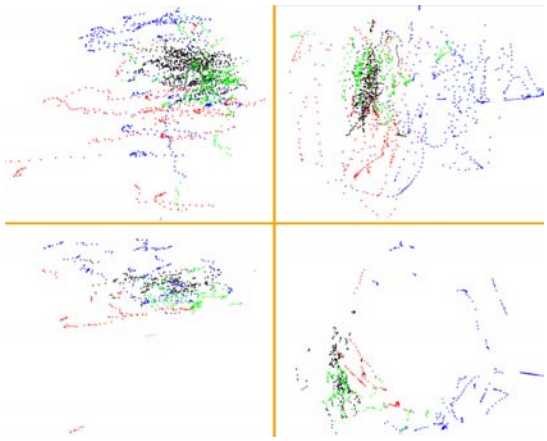

Figure 5: *Comparisons between 2D Phoneme-PCA maps and 2D Phoneme-Isomaps.*

phoneme-Isomaps are not directly suitable for interactively browsing and selecting facial motion-frames. A Gaussian kernel point-rendering visualizes the Isomaps, where pixels accumulate the Gaussian distributions centered at each embedded location. Pixel colors are proportional to the probability of a corresponding motion-frame representing the phoneme. In this way, we generated a phoneme-Isomap image for each phoneme-Isomap (Fig. 6).

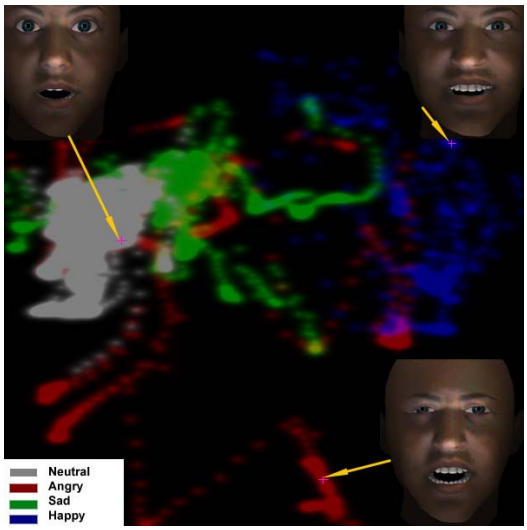

Figure 6: *Illustration of a 2D expressive phoneme-Isomap for phoneme /ay/.*

A 2D Delaunay triangulation algorithm is applied to the embedded 2D Isomap coordinates of each phoneme-Isomap to produce a triangulation network. Each vertex of these triangles corresponds to an embedded phoneme-Isomap point (a motion-frame in the phoneme cluster). These triangles cover

most of the points in the phoneme-Isomap image without overlap (some points around the image boundary are not covered by the triangulation network). Therefore, when a point in the Phoneme Isomaps is picked, its 2D position is mapped back to the 2D embedded Isomap coordinate system, then the mapped position determines the unique covering triangle. The barycentric interpolation is used to interpolate three vertices (motion-frames) of the covering triangle to generate a new motion-frame (corresponding to the picked point). A phoneme-Isomap image is a visualized representation of a continuous space of recorded facial configurations for one specific phoneme (Fig. 6). The phoneme-Isomap image of the /ay/ phoneme is shown in Fig. 6. Note that these phoneme-Isomap images and their mapping/triangulation information were precomputed and stored for later use. Based on the above interpolated motion frame (for any picked point), a 3D face model is deformed correspondingly. A feature point based mesh deformation approach [KGT00] is used for this rapid deformation.

5. Motion Editing

The captured facial motion database is composed of hundreds of thousands of motion capture frames, and it is challenging to manage and edit these huge data. The phoneme-Isomap images allow users to edit such huge facial motion data. Users can interactively create and add new motion nodes into the facial motion database.

As described in Section 3, each motion node is a sequence of motion capture frames of one specific phoneme in their recorded order. It is visualized as a directed trajectory (curve) in phoneme-Isomap images. Since each point on the trajectory represents a specific facial configuration (see Fig. 6), and the image color behind a motion-node trajectory represents the emotion category of the motion node, users can intuitively and conveniently inspect any frame in the motion node (a point on the trajectory) as follows: when users click any point on the trajectory, its corresponding 3D face deformation is interactively displayed in a preview window. Besides offering motion frame preview, our system can be straightforwardly extended to handle previewing "expressive facial motion clips": if users select one motion node in a phoneme-Isomap, a clip preview window can show the animation of the corresponding motion node (facial motion segment).

On the other side, if contaminated motion nodes are found, users can choose to select and delete these motion nodes from the database, so that the motion synthesis algorithm (Section 6) could avoid the risk of being trapped into these contaminated motion nodes. Based on existing motion nodes and their corresponding trajectories in phoneme-Isomap images, users can create new motion nodes by drawing free-form 2D trajectories (each continuous trajectory corresponds to a new motion node). In this way, users can expand the facial motion database.

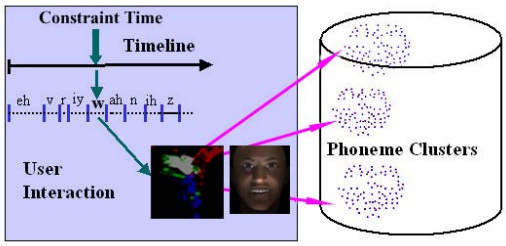

Figure 7: *Illustration of how to specify a motion-node constraint via the phoneme-Isomap interface. When users want to specify a specific motion node for expressing a particular phoneme utterance, its corresponding phoneme-Isomaps are automatically loaded. Then, users can interact with the system to specify a motion-node constraint for this constrained phoneme.*

6. Motion Synthesis

In this section, we describe how our motion synthesis algorithm synthesizes corresponding facial motion, given a novel phoneme sequence and its emotion specifications as input. The system is fully automatic while providing optional intuitive controls: users can specify a motion-node constraint for any phoneme utterance ("hard constraints") via the above phoneme-Isomap interface, and our algorithm will automatically regard the emotion modifiers as "soft constraints". Under these hard and soft constraints, our algorithm searches for a best-matched path of motion nodes from the processed facial motion node database by minimizing a cost function using a constrained dynamic programming technique.

6.1. Specify Motion-Node Constraints

Users interactively browse phoneme-Isomap images to specify motion-node constraints and tie them to a specific phoneme utterance's expression. We refer to this time as *a constrained time* and its corresponding phoneme as *a constrained phoneme*. Phoneme timing is included in the preprocessed phrase (phoneme) transcript, so phoneme-Isomaps are automatically loaded once a constrained phoneme is selected (Fig. 7).

To guide users in identifying and selecting proper motion nodes, our system automatically highlights recommended motion nodes and their picking points. Assuming a motion node path $o_1, o_2, \ldots, o_k$ is obtained by our automatic motion-path search algorithm (the follow-up Section 6.2 details this algorithm), users want to specify a motion-node constraint for a constrained time T_c (assume its corresponding constrained phoneme is P_c and its motion-frame at T_c is F_c, called *current selected frame*). The constrained time T_c is first divided by the duration of the constrained phoneme P_c to calculate its relative time $t_c (0 \leq t_c \leq 1)$. Then, for each

motion node in the phoneme cluster, the system highlights one of its motion frames whose relative time property is the closest to current relative time t_c. We refer to these motion frames as *time-correct motion frames*.

As mentioned in Section 3, the specific triphone context of each motion node was also retained. By matching the triphone context of the constrained phoneme with those of existing motion nodes in the phoneme cluster of P_c, our system identifies and highlights the motion nodes in the phoneme cluster that have the same triphone context as the constrained phoneme (termed *context-correct motion nodes*). For example, in Fig. 7, the current constrained phoneme is /w/, and its triphone context is [/iy/, /w/, /ah/], so the system will identify the motion nodes of the /w/ phoneme cluster that have the triphone context [/iy/, /w/, /ah/] as the context-correct motion nodes. In this way, by picking their representative time-correct motion frames, users can choose one of those motion nodes as a motion-node constraint for P_c. This motion node constraint is imposed per phoneme utterance, in other words, if one specific phoneme appears multiple times in a phoneme input sequence, users can specify different motion-node constraints for them. Figure 8 shows a snapshot of phoneme-Isomap highlights for specifying motion-node constraints. Note that the background phoneme-Isomap image is always the same for a specific phoneme, but these highlighting symbols (Fig. 8) are related to current relative time t_c and current triphone context. So, these markers are changed over time (even for the same phoneme).

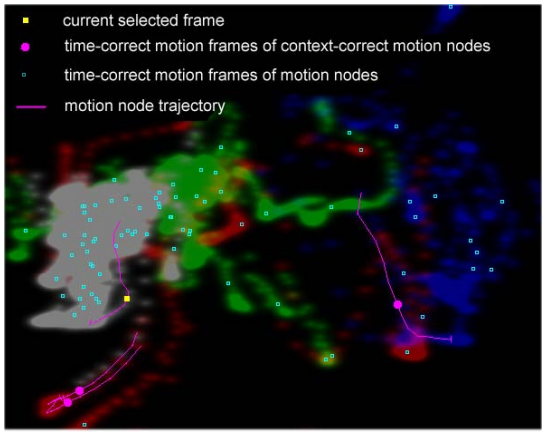

Figure 8: *A snapshot of phoneme-Isomap highlights for specifying motion-node constraints.*

6.2. Search for the Optimal Concatenations

We can formalize the motion-node path search problem as follows: Given a novel phomeme sequence input $\Psi = (P_1, P_2, \cdots, P_T)$ and its emotion modifiers $\Theta = (E_i, E_2, \cdots, E_T)$ (E_i only can be one of four possible values: neutral, angry, sad and happy), and optional motion-node constraints $\Phi = (C_{t_1} = o_{i_1}, C_{t_2} = o_{i_2}, \cdots, C_{t_k} = o_{i_k}, t_i \neq t_j)$, we want to search for a best-matched motion-node path $\Gamma^* = (o^*_{\rho_1}, o^*_{\rho_2}, \cdots, o^*_{\rho_T})$ that minimizes a cost function $COST(\Psi, \Theta, \Phi, \Gamma^*)$. Here o_i represents a motion node with index i.

To make the definition of the above cost function clear, we first leave out the constraint parameter Φ and define a plain version $COST(\Psi, \Theta, \Gamma)$ without motion-node constraints. We will describe how the constraint parameter Φ affect the cost function and the search process later in this section. The cost function $COST(\Psi, \Theta, \Gamma)$ is the accumulated summation of Transition Cost $TC(o_{\rho_i}, o_{\rho_{i+1}})$, Observation Cost $OC(P_i, o_{\rho_i})$, and Emotion Mismatch Penalty $EMP(E_i, o_{\rho_i})$, as described in Equation 1. Here Transition Cost $TC(o_{\rho_i}, o_{\rho_{i+1}})$ represents the smoothness of the transition from one motion node o_{ρ_i} to the other motion node $o_{\rho_{i+1}}$, and Observation Cost $OC(P_i, o_{\rho_i})$ measures the goodness of a motion node o_{ρ_i} for expressing a given phoneme P_i. To compute $TC(o_{\rho_i}, o_{\rho_{i+1}})$, Direct Smoothing Cost $DSC(o_{\rho_i}, pre(o_{\rho_{i+1}}))$ and Position Velocity Cost $PVC(o_{\rho_i}, o_{\rho_{i+1}})$ are weight added. If the emotion label of a motion node o_{ρ_i} is same as the specified emotion modifier E_i, we set the emotion mismatch penalty $EMP(E_i, o_{\rho_i})$ to zero, otherwise it is set to a constant penalty value.

$$COST(\Psi, \Theta, \Gamma) = \sum_{i=1}^{T-1} TC(o_{\rho_i}, o_{\rho_{i+1}}) + \sum_{i=1}^{T} (OC(P_i, o_{\rho_i}) + EMP(E_i, o_{\rho_i})) \quad (1)$$

Based on the above cost definitions, we use the dynamic programming algorithm to search for the best-matched motion-node sequence $\Gamma^*(o^*_{\rho_1}, o^*_{\rho_2}, \cdots, o^*_{\rho_T})$. Assume there are total N motion nodes in the processed motion node database. This search algorithm can be described as follows:
(1) Initialization (for $1 \leq i \leq N$):

$$\varphi_1(i) = OC(P_1, o_i) + EMP(E_1, o_i) \quad (2)$$

(2) Recursion (for $1 \leq j \leq N; 2 \leq t \leq T$):

$$\varphi_t(j) = \min_i \{\varphi_{t-1}(i) + TC(o_i, o_j) + OC(P_t, o_j) + EMP(E_t, o_j)\} \quad (3)$$

$$\chi_t(j) = \arg\min_i \{\varphi_{t-1}(i) + TC(o_i, o_j) + OC(P_t, o_j) + EMP(E_t, o_j)\} \quad (4)$$

(3) Termination:

$$COST^* = \min_i \{\varphi_T(i)\} \quad (5)$$

$$\rho^*_T = \arg\min_i \{\varphi_T(i)\} \quad (6)$$

(4) Recover path by backtracking (t from $T-1$ to 1):

$$\rho_t^* = \chi_{t+1}(\rho_{t+1}^*) \tag{7}$$

In this way, we can find the best-matched motion-node path $\Gamma^* = (o^*_{\rho_1}, o^*_{\rho_2}, \cdots, o^*_{\rho_T})$. The time complexity of the above search algorithm is $\Theta(N^2 * T)$, here N is the number of motion nodes in the database and T is the length of input phonemes.

Now we describe how the specified motion-node constraints $\Phi = (C_{t_1} = o_{i_1}, C_{t_2} = o_{i_2}, \cdots, C_{t_k} = o_{i_k}, t_i \neq t_j)$ affect the above search algorithm to guarantee that the searched motion-node path passes through the specified motion nodes at specified times. The constraints affect the search process by blocking the chances of other motion nodes (except the specified ones) at certain recursion time. Eq. 3-4 in the above search algorithm are replaced with the following new equations (8-10).

$$\varphi_t(j) = \min_i\{\varphi_{t-1}(i) + TC(o_i, o_j) + OC(P_t, o_j) + EMP(E_t, o_j) + B_t(j)\} \tag{8}$$

$$\chi_t(j) = \arg\min_i\{\varphi_{t-1}(i) + TC(o_i, o_j) + OC(P_t, o_j) + EMP(E_t, o_j) + B_t(j)\} \tag{9}$$

$$B_t(j) = \begin{cases} 0 & \text{if } \exists m, t_m = t \text{ and } j = i_m \\ HugePenalty & \text{otherwise} \end{cases} \tag{10}$$

Given the optimal motion-node path Γ^*, we concatenate its motion nodes by smoothing their neighboring boundaries and transforming facial motions of the motion nodes from their retained PCA space to markers' 3D space. Finally, we transfer the synthesized marker motion sequence onto specific 3D face models.

7. Results and Evaluations

We developed the eFASE system using VC++ that runs on the MS Windows XP system. Fig. 9 shows a snapshot of the running eFASE system. Table 1 illustrates an example of a phoneme input file and an emotion specification file.

We conducted a running time analysis on the eFASE system. The computer used is a Dell Dimension 4550 PC (Windows XP, 1GHz Memory, Intel 2.66GHz Processor). Table 2 encloses the running time of some example inputs. As mentioned in Section 6.2, the motion node searching part (the most time-consuming part of the eFASE system) has a time complexity of $\Theta(N^2 * T)$ that is linear to the length of input phonemes (assuming N is a fixed value for a specific database). The computing time enclosed in the Table 2 is approximately matched with this analysis.

We also compared the synthesized expressive facial motion

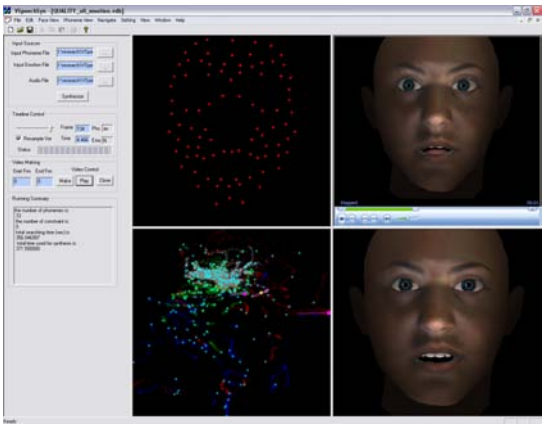

Figure 9: *A snapshot of the running eFASE system. The left is a basic control panel, and the right panel encloses four working windows: a synthesized motion window (top-left), a video playback window (top-right), a phoneme-Isomap interaction window (bottom-left), and a face preview window (bottom-right).*

0.122401 pau	2.6 angry
0.24798 ay	16.6383 sad
0.328068 ae	
0.457130 m	
0.736070 n	
...	

Table 1: *An example of an aligned phoneme input file (left) and an emotion modifier file (right). Its phrase is " I am not happy...". Here the emotion of the starting 2.6 second is angry, and the emotion from #2.6 second to #16.6383 second is sadness.*

phrases (number of phonemes)	time (second)
"I know you meant it" (14)	137.67
"And so you just abandoned them?" (24)	192.86
"Please go on, because Jeff's father has no idea" (33)	371.50
"It is a fact that long words are difficult to articulate unless you concentrate" (63)	518.34

Table 2: *Running time of synthesis of some example phrases. Here the computer used is a Dell Dimension 4550 PC (Windows XP, 1GHz Memory, Intel 2.66GHz Processor).*

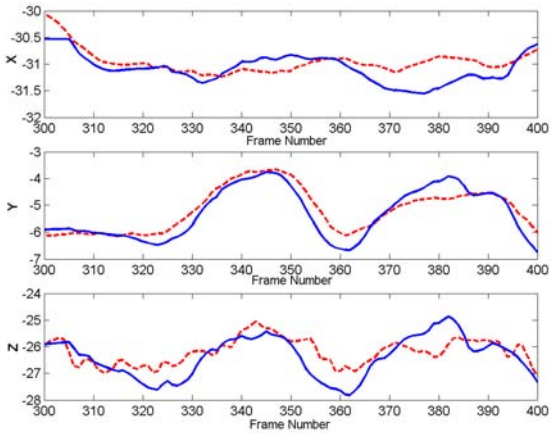

Figure 10: *A part of marker (#48 marker) trajectory of the sad sentence "Please go on, because Jeff's father has no idea of how the things became so horrible." The dashed line is the groundtruth trajectory and the solid line is the synthesized trajectory.*

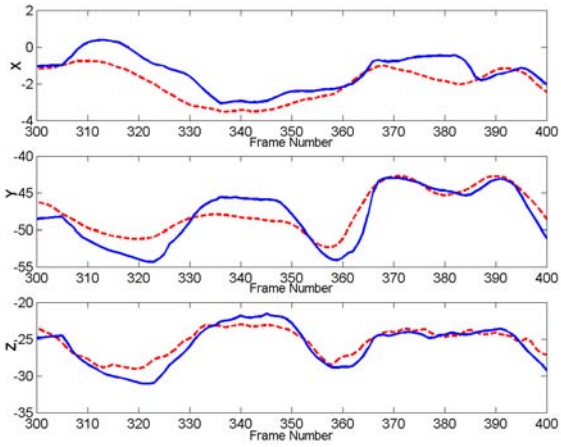

Figure 11: *A part of marker (#79 marker) trajectory of the sad sentence "Please go on, because Jeff's father has no idea of how the things became so horrible." The dashed line is the ground truth trajectory and the solid line is the synthesized trajectory.*

with corresponding captured motion. Twelve additional sentences were exclusively used for test comparisons. One of these sentences was "*Please go on, because Jeff's father has no idea of how the things became so horrible.*" with sad expression. We chose a right cheek marker (#48 marker) in an expression-active area and a lower lip marker (#79 marker) in a speech-active area for the comparisons (Fig. 2). We plotted a part of the synthesized sequence and ground truth motion for these marker trajectory comparisons. Fig. 10 is for #48 marker (the right cheek marker) and Fig. 11 is for #79 marker (the lower lip marker). We found that the trajectories of the synthesized motions are quite close to the actual motions captured from the actress. Notice that the synthesized motions for these comparisons (Fig. 10 and 11) were automatically generated without any manual intervention (i.e. without the use of motion-node constraints). We also synthesized numerous expressive facial animations using novel recorded and archival speech.

8. Discussion and Conclusions

We present a data-driven expressive facial animation synthesis and editing system (eFASE) with intuitive phoneme-level control. Users control the facial motion synthesis process by specifying emotion modifiers and expressions for certain phoneme utterances via novel 2D expressive phoneme-Isomaps. This system employs a constrained dynamic programming algorithm that satisfies hard constraints (motion-node constraints) and soft constraints (specified emotions). Objective trajectory comparisons between synthesized facial motion and captured motion, and novel synthesis experiments, demonstrate that the eFASE system is effective for producing realistic expressive facial animations.

This method introduces the Isomap framework [TSL00] for generating intuitive low-dimensional manifolds for each phoneme cluster. The advantage of the Isomap (over PCA, for example) is that it leads to a better projection of motion frames with different emotions, and it makes browsing and editing expressive motion sequences (and frames) more intuitive and convenient. An interactive and intuitive way of browsing and selecting among the large number of phoneme variations is itself a challenging problem for facial animation research.

As this is a new approach to facial animation synthesis and editing, several issues require further investigations. The quality of novel motion synthesis depends on constructing a large facial motion database with accurate motion and phoneme alignment. Building this database takes care and time; integrated tools could improve this process immensely. Current system offers a novel way to interactively create new motion nodes from phoneme-Isomaps, extensions of facial animation editing techniques [JTDP03, ZLGS03] that automatically modify the whole face in response to a local user change could be another promising method to further flexibly expand the facial motion database.

Current system cannot be used for real-time applications. Optimizations could further improve efficiency by reducing the size of the facial motion database through clustering methods. We are aware that subjective evaluation would be helpful to quantify and improve our system, and we plan to look into it in the future. Emotion intensity control that is

absent in current system is another good direction for future improvement.

The motions of the silence phoneme (the /pau/ phoneme in the *Festival* system) are not modeled. This phoneme and other non-speaking animations (e.g. yawning) need to be represented as motion nodes to allow more flexibility and personified realism. Lastly, there are more open questions, such as whether combining the speaking styles of different actors into one facial motion database would result in providing a greater range of motions and expressions, or if such a combination would muddle the motion-frame sequencing and expressiveness, or whether exploiting different weights for markers to guide the coherence of perceptual saliency could improve results.

Acknowledgements

This research has been funded by the Integrated Media System Center at University of Southern California (USC), a 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 author(s) and do not necessarily reflect those of the National Science Foundation. Special Thanks go to J.P. Lewis and Yizhou Yu for thoughtful suggestions, Pamela Fox for face model preparation, Joy Nash, Murtaza Bulut, and Carlos Busso for facial motion data capture and processing.

References

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating faces in images and video. *Computer Graphics Forum 22*, 3 (2003).

[BCS97] BREGLER C., COVELL M., SLANEY M.: Video rewrite: Driving visual speech with audio. *Proc. of ACM SIGGRAPH'97* (1997), 353–360.

[BDNN05] BUSSO C., DENG Z., NEUMANN U., NARAYANAN S.: Natural head motion synthesis driven by acoustic prosody features. *Computer Animation and Virtual Worlds 16*, 3-4 (July 2005), 283–290.

[Bra99] BRAND M.: Voice pupperty. *Proc. of ACM SIGGRAPH'99* (1999), 21–28.

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. *Proc. of ACM SIGGRAPH'99* (1999), 187–194.

[CB05] CHUANG E., BREGLER C.: Moodswings: Expressive speech animation. *ACM Trans. on Graph. 24*, 2 (2005).

[CDB02] CHUANG E. S., DESHPANDE H., BREGLER C.: Facial expression space learning. In *Proc. of Pacific Graphics'2002* (2002), pp. 68–76.

[CFKP04] CAO Y., FALOUTSOS P., KOHLER E., PIGHIN F.: Real-time speech motion synthesis from recorded motions. In *Proc. of Symposium on Computer Animation* (2004), pp. 345–353.

[CG00] COSATTO E., GRAF H. P.: Audio-visual unit selection for the synthesis of photo-realistic talking-heads. In *Proc. of ICME* (2000), pp. 619–622.

[CM93] COHEN M. M., MASSARO D. W.: Modeling coarticulation in synthetic visual speech. *Models and Techniques in Computer Animation, Springer Verlag* (1993), 139–156.

[CXH03] CHAI J., XIAO J., HODGINS J.: Vision-based control of 3d facial animation. In *Proc. of Symposium on Computer Animation* (2003), ACM Press, pp. 193–206.

[DCFN06] DENG Z., CHIANG P. Y., FOX P., NEUMANN U.: Animating blendshape faces by cross-mapping motion capture data. In *Proc. of ACM SIGGGRAPH Symposium on Interactive 3D Graphics and Games* (2006), pp. 43–48.

[DLN05] DENG Z., LEWIS J. P., NEUMANN U.: Synthesizing speech animation by learning compact speech co-articulation models. In *Proc. of Computer Graphics International* (2005), pp. 19–25.

[DNL*06] DENG Z., NEUMANN U., LEWIS J. P., KIM T. Y., BULUT M., NARAYANAN S.: Expressive facial animation synthesis by learning speech co-articulations and expression spaces. *IEEE Trans. Vis. Graph. 12*, 6 (2006).

[EGP02] EZZAT T., GEIGER G., POGGIO T.: Trainable videorealistic speech animation. *ACM Trans. Graph.* (2002), 388–398.

[fes04] http://www.cstr.ed.ac.uk/projects/festival/, 2004.

[GGW*98] GUENTER B., GRIMM C., WOOD D., MALVAR H., PIGHIN F.: Making faces. *Proc.of ACM SIGGRAPH'98* (1998), 55–66.

[JTDP03] JOSHI P., TIEN W. C., DESBRUN M., PIGHIN F.: Learning controls for blend shape based realistic facial animation. In *Proc. of Symposium on Computer animation* (2003), pp. 187–192.

[KGT00] KSHIRSAGAR S., GARCHERY S., THALMANN N. M.: Feature point based mesh deformation applied to mpeg-4 facial animation. In *Proc. Deform'2000, Workshop on Virtual Humans by IFIP Working Group 5.10* (November 2000), pp. 23–34.

[KHS01] KÄHLER K., HABER J., SEIDEL H. P.: Geometry-based muscle modeling for facial animation. In *Proc. of Graphics Interface'2001* (2001).

[KP05] KING S. A., PARENT R. E.: Creating speech-synchronized animation. *IEEE Trans. Vis. Graph. 11*, 3 (2005), 341–352.

[KT03] KSHIRSAGAR S., THALMANN N. M.: Visyllable based speech animation. *Computer Graphics Forum 22*, 3 (2003).

[Lew91] LEWIS J. P.: Automated lip-sync: Background and techniques. *Journal of Visualization and Computer Animation* (1991), 118–122.

[LTW95] LEE Y. C., TERZOPOULOS D., WATERS K.: Realistic modeling for facial animation. *Proc. of ACM SIGGRAPH'95* (1995), 55–62.

[MCP*05] MA J., COLE R., PELLOM B., WARD W., WISE B.: Accurate visible speech synthesis based on concatenating variable length motion capture data. *IEEE Transaction on Visualization and Computer Graphics (online)* (2005).

[NN01] NOH J. Y., NEUMANN U.: Expression cloning. *Proc. of ACM SIGGRAPH'01* (2001), 277–288.

[Pel91] PELACHAUD C.: Communication and coarticulation in facial animation. *Ph.D. Thesis, Univ. of Pennsylvania* (1991).

[PHL*98] PIGHIN F., HECKER J., LISCHINSKI D., SZELISKI R., SALESIN D. H.: Synthesizing realistic facial expressions from photographs. *Proc. of ACM SIGGRAPH'98* (1998), 75–84.

[PKC*03] PYUN H., KIM Y., CHAE W., KANG H. W., SHIN S. Y.: An example-based approach for facial expression cloning. In *Proc. of Symposium on Computer Animation* (2003), pp. 167–176.

[PW96] PARKE F. I., WATERS K.: *Computer Facial Animation*. A K Peters, Wellesley, Massachusets, 1996.

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3 (2004), 514–521.

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (2005), 417–425.

[TSL00] TENENBAUM J., SILVA V. D., LANGFORD J.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2333.

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3 (2005), 426–433.

[WF95] WATERS K., FRISBLE J.: A coordinated muscle model for speech animation. *Proc. of Graphics Interface'95* (1995), 163–170.

[Wil90] WILLIAMS L.: Performance-driven facial animation. In *Proc. of ACM SIGGRAPH '90* (1990), ACM Press, pp. 235–242.

[ZLGS03] ZHANG Q., LIU Z., GUO B., SHUM H.: Geometry-driven photorealistic facial expression synthesis. In *Proc. of Symposium on Computer Animation* (2003), pp. 177–186.

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph. 23*, 3 (2004), 548–558.