To appear in the Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2006 (I3DG), March 2006

Animating Blendshape Faces by Cross-Mapping Motion Capture Data

Zhigang Deng* USC
Pei-Ying Chiang USC
Pamela Fox USC
Ulrich Neumann † USC

Abstract

Animating 3D faces to achieve compelling realism is a challenging task in the entertainment industry. Previously proposed face transfer approaches generally require a high-quality animated source face in order to transfer its motion to new 3D faces. In this work, we present a semi-automatic technique to directly animate popularized 3D blendshape face models by mapping facial motion capture data spaces to 3D blendshape face spaces. After sparse markers on the face of a human subject are captured by motion capture systems while a video camera is simultaneously used to record his/her front face, then we carefully select a few motion capture frames and accompanying video frames as *reference mocap-video pairs*. Users manually tune blendshape weights to perceptually match the animated blendshape face models with reference facial images (the reference mocap-video pairs) in order to create *reference mocap-weight pairs*. Finally, the Radial Basis Function (RBF) regression technique is used to map any new facial motion capture frame to blendshape weights based on the reference mocap-weight pairs. Our results demonstrate that this technique is efficient to animate blendshape face models, while offering its generality and flexiblity.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

Keywords: Facial Animation, Blend Shape Models, Cross-Mapping, Motion Capture Data, Radial Basis Functions

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. Furthermore, they have to repeat a similiar process on different face models. How to animate various face models to achieve compelling realism with as little manual intervention as possible has been a challenging research topic for computer animation researchers.

The blendshapes (shape interpolation) method is a popular approach to computer facial animation. For example, the "Gollum" model used in *Lord of the Rings* movies has a total of 946 blendshape controls [Lewis et al. 2005]. The blendshapes method is used by animators so frequently because of its intuitive and flexible controls. However, the cost of blendshape face animation is considerable. On top of the considerable time taken for the original blendshape sculpting that must be done once for each model, still more time is taken up by the tedious task of animating pre-designed blendshape models that must be done for each new dialogue.

Face transfer techniques [Noh and Neumann 2001; Pyun et al. 2003] reused existing facial animation by transferring source facial animation to target face models with little manual intervention, but these techniques require a high-quality animated source face. Recent face transfer work [Vlasic et al. 2005] successfully demonstrated how to transfer a recorded face video on to 3D face models. However, one of major limitations of all the above approaches is that target faces are currently limited to non-blendshape face models, and these approaches can not be directly applied to a pre-designed blendshape face model without considerable efforts. Pyun et al. [2003] and Ma et al. [2004] require animators to customize a proper blendshape face model by sculpting a blendshape basis based on each key facial configuration chosen from source facial animation sequences. However, these techniques cannot be directly applied to blendshape face models that were sculpted in advance (often with great effort, like the "Gollum" model).

Motion capture has proven to be a feasible and robust approach to acquire high-quality human motion data including facial motion, and it is in popular use by the current entertainment industry [Scott 2003]. Hence, directly linking facial motion capture data with the popular blendshape face method has widespread applications in the entertainment industry and could greatly alleviate the pains of making compelling facial animation.

Animating blendshape faces from motion capture data is essentially a mapping from a motion capture data space to a blendshape weight space. Since multiple blendshape weight combinations could achieve the same facial configuration, this mapping generally is a one-to-many mapping that imposes challenges to full-automatic approaches. In this work, we present a semi-automatic approach to animate arbitrary blendshape face models, and we formalize it as a scattered data interpolation problem [Nielson 1993]: based on a few carefully chosen reference correspondences between blendshape weights and motion capture frames, blendshape weights of new motion capture frames are computed using Radial Basis Function (RBF) regression. Figure 1 illustrates the schematic overview of this approach.

As illustrated in Figure 1, in the first stage (data capture), a motion capture system and a video camera are simultaneously used to record the facial motions of a human subject. The audio recordings from the two systems are misaligned with a fixed time-shift because of slight differences in start time of recording. The manual alignment of these two audio recordings results in strict alignments between mocap frames and video frames. In the second stage, we carefully select a few of the aligned pairs as *reference mocap-video pairs* so that they cover the spanned space of visemes and emotions as completely as possible. In the third stage, motion capture data are reduced to a low dimensional space by Principal Component Analysis (PCA). Meanwhile, based on the selected reference video frames (face snapshots), animators manually tune the weights of the blendshape face model to perceptually match the model and the reference images, which creates surpervised correspondences between the PCA coefficients of motion capture frames and the weights of the blendshape face model (referred to as *mocap-weight pairs*). Taking the reference mocap-weight pairs as training exam-

*Email: zdeng@graphics.usc.edu
†Email: uneumann@graphics.usc.edu

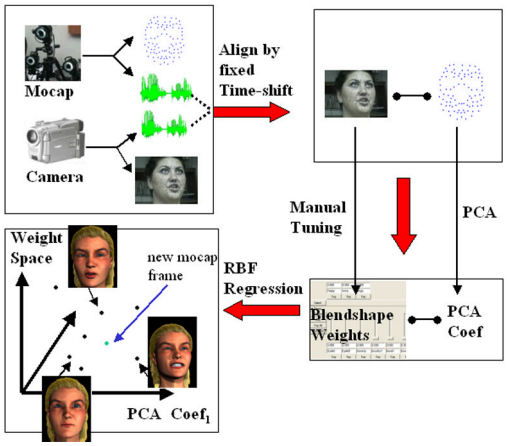

Figure 1: Schematic overview of this work. It consists of four stages: data capture, creation of mocap-video pairs, creation of mocap-weight pairs, and RBF regression.

ples, the Radial Basis Function (RBF) regression technique is used to automatically compute blendshape weights for new input motion capture frames (the fourth stage in Fig. 1).

In this paper, we show that directly animating blendshape face models from facial motion capture data can be achieved by formalizing it as a scattered data interpolation problem. The introduced mocap-video pairs enable animators to make a direct link between mocap frames and blendshape weights. This technique can save considerable efforts when dealing with a huge motion capture database, because it only requires animators to do manual tuning on a small number of selected examples.

The remainder of the paper is organized as follows: Section 2 reviews recent work in face transfer and blendshape animation, Section 3 describes the details of data acquisition and process, Section 4 describes the selection of reference mocap frames from database, Section 5 describes the construction of mocap-weight pairs, Section 6 details the computing of blendshape weights using RBF regression, given new input motion capture frames, Section 7 describes experimental results, and finally, Section 8 concludes this paper.

2 Related Work

In this section, we will specifically review recent work related to face transfer and blendshape facial animation. More topics on facial animation and modeling can be found in the facial animation book [Parke and Waters 1996].

The expression cloning technique [Noh and Neumann 2001] transfers vertex displacements from a source face model to target face models that may have different geometric proportions and mesh structure. Generally, it requires little manual intervention, because it uses heuristic rules to search for vertex correspondences. However, it requires a high-quality animated face as its source animation. The general deformation transfer technique [Sumner and Popović 2004] can be used for retargetting facial motion from one model to another model, but it shares the same problem of the expression cloning technique [Noh and Neumann 2001]. Pyun et al. [2003] and Ma et al. [2004] transfer facial animation using an example-based approach. Essentially these approaches require animators to customize a proper blendshape face model by sculpting a blendshape basis based on each key facial configuration chosen from source facial animation sequences. Hence, this technique can not be directly applied to other pre-designed blendshape models without considerable efforts.

Recently Vlasic et al. [2005] presented a novel framework to transfer facial motion in video to other 2D or 3D faces by learning a multilinear model of 3D face meshes. In their work, the learned multilinear face models are controlled via intuitive attribute parameters, such as identity and expression. Varying one attribute parameter (e.g. identity) while keeping other attributes constant can transfer the facial motion of one model to another. It has yet to be demonstrated whether this face transfer technique can be successfully used in animating the blendshape face models that are preferred by animators.

Pose Space Deformation (PSD) presented by Lewis et al. [2000] provides a general framework for example-based deformation. In their work, the deformation of a surface is treated as a function of some set of abstract parameters, such as {*smile, raise-eyebrow,...*}, and new surface deformation (pose) is interpolated by a scattered data interpolation. In some respects, our work shares similiarities with the PSD, like the use of scattered data interpolation. However, the major distinction of this work is that it treats abstract or high-level parameters (blendshape weights, e.g. *raise-eyebrow*) as a function of low level parameters, such as PCA coefficients. It is the inverse of what the PSD does. Additionally, it has not been verified yet that PSD can be directly used to map motion capture data to animate blendshape faces.

Blanz and Vetter [1999] presented a novel technique to learn a morphable face model by applying Principal Component Analysis (PCA) to vector representations of the shape and texture of 3D face examples. These extracted principal components correspond to prototype faces (blendshape basis), and new faces can be successfully modeled by adjusting blending weights of the prototype faces, manually or automatically. Furthermore, on top of the morphable face model, Blanz et al. [2003] reanimate faces in images/video by first fitting the morphable face model to 2D faces and then transferring facial motion to animate the fitted 3D faces. Lewis et al. [2005] presented an efficient way to reduce the blendshape interference problem by selected motion attenuation. After animators move blendshape slider values, their approach will automatically re-adjust blendshape slider values by mimicing new movement while moving some tagged vertices as little as possible.

Choe and Ko [2001] presented a technique to animate blendshape face models composed of hand-generated muscle actuation base, by iteratively adjusting muscle actuation base and analyzed weights. Sifakis et al. [2005] use nonlinear finite element methods to determine accurate muscle actuations from motions of sparse facial markers. In their work, an anatomically accurate model of facial musculature, passive tissue, and underlying skeleton structure is used. Their work demonstrated the success of the techniques on muscle actuation based blendshape models or anatomically accurate face models, but it is not clear whether these muscle-based approaches can be extended to handle general blendshape face models that are not based on muscle actuation.

Performance driven facial animation was introduced by Williams [1990] that tracks facial motion of real actors (performance) and drives other 3D face models accordingly. Recent improved performance driven facial animation techniques focused on more robust tracking and retargetting algorithms [Li et al. 1993; Essa et al. 1996; Decarlo and Metaxas 1996; Pighin et al. 1999; Noh et al. 2002; Chai et al. 2003]. These techniques require robust vision-tracking algorithms, and the success of directly applying these techniques to pre-designed blendshape face models has not

yet been demonstrated.

Data-driven speech animation synthesis can be regarded as generalized face transfer [Bregler et al. 1997; Brand 1999; Ezzat et al. 2002; Kshirsagar and Thalmann 2003; Cao et al. 2004; Ma et al. 2004; Deng et al. 2005]. Most of these approaches first record facial motion of speaking subjects. Then, given new speech input, they recombine recorded facial motion frames or sample from learned parametric models to synthesize new facial motion. Hence, they "transfer" pre-recorded motion frames for new utterances, although generally this transfer happens with respect to the same face.

3 Data Acquisition

A VICON motion capture system with camera rigs (top panel of Fig 2) with a 120Hz sampling rate, is used to capture expressive facial motion data of an actress speaking in a normal speed, with 102 markers on her face. In this recording, four basic emotions (neutral, happiness, anger, and sadness) were considered. The actress was asked to speak the sentences with full intensity of emotions. The markers' motion and aligned audio were recorded by the system simultaneously. At the same time, we used another video camera to record the front of her face (Fig. 1 and the bottom panel of Fig. 2).

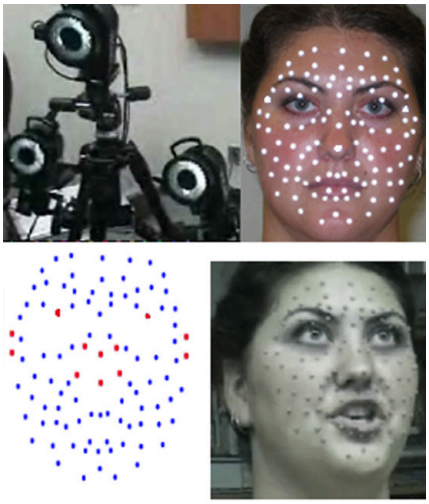

Figure 2: Illustration of facial motion data acquisition. The top panel shows the camera rigs of the motion capture system and the used facial marker layout. The bottom panel shows motion capture data and a snapshot of recorded face video. Due to marker tracking erros at some frames, red markers in the bottom-left panel were not used in this work.

We then further normalized the captured 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 was used to calculate head motion [Stegmann and Gomez 2002; Busso et al. 2005]. A neutral pose was chosen as a reference frame that is packed into a 102×3 matrix, y. For each motion capture frame, a matrix x_i was created using the same marker order as the reference. After that, the *Singular Value Decomposition*(SVD), UDV^T, of matrix $y^T x_i$ was calculated. Finally, the product of VU^T gave the rotation matrix, R.

$$y^T x_i = UDV^T \tag{1}$$

$$R = VU^T \tag{2}$$

Since both video and its audio were recorded by the video camera, they were perfectly aligned. Similiarly, facial motion capture data were perfectly aligned with the audio recorded by the motion capture system. However, because of slightly different recording-start times, audio recorded by the motion capture system had a fixed-time shift difference with the audio recorded by the video camera, and the same time-shift difference applies for motion capture frames and the video frames. After we manually aligned these two audio segments and calculated the fixed-time shift, shifted facial motion capture frames became strictly aligned with the recorded video frames. Figure 2 illustrates snapshots of this data acquisition process. Due to marker tracking erros at some frames, red markers in the bottom-left panel were not used in this work.

4 Select Reference Frames

In this section, we describe the selection of reference motion capture frames from recorded motion capture segments. For motion capture data, the FESTIVAL system [FESTIVAL] was used to perform phoneme-alignment on their accompanying audio in order to align each phoneme with its corresponding motion capture segments. This alignment was done by feeding audio and its accompanying text scripts into the speech recognition program in a force-alignment mode.

Based on the above phoneme-alignment results, we manually selected one reference motion capture frame C_i (generally, the middle frame) from a phoneme-associated motion capture segment, for each possible combination of visemes (Table 1) and expressions: 15 visemes*4 expressions=60. A *viseme* is the visual equivalent of a phoneme or unit of sound in spoken language. Furthermore, we selected addtional 40 frames for extreme expressive poses (10 frames for each expression). These selected 100 motion capture frames $\{C_i\}$ with their aligned video frames $\{V_i\}$ are referred as *reference mocap-video pairs*, $\{\langle C_i, V_i\rangle\}$.

We assume most of current facial animation applications can be approximately expressed in terms of speech and expressions. Another consideration is that the space spanned by these reference mocap frames should be as balanced and complete as possible. These are major reasons that we chose representative expressive and viseme frames as the reference mocap-video pairs.

/pau/	/b/, /p/, /m/	/k/, /g/, /ng/
/ae/, /ax/, /ah/, /aa/	/f/, /v/	/ch/, /sh/, /jh/
/ao/, /y/, /iy/, /ih/, /ay/, /aw/	/ow/, /oy/	/n/, /d/, /t/, /l/
/ey/, /eh/, /el/, /em/, /en/, /er/	/th/, /dh/	/s/, /z/, /zh/
/r/	/w/, /uw/, /uh/	/hh/

Table 1: Scheme of grouping phonemes into the 15 visemes used in this work.

5 Mocap-Weight Pairs

In this section, we describe the construction of reference mocap-weight pairs $\{\langle CF_i, W_i\rangle\}$ based on the reference mocap-video pairs $\{\langle C_i, V_i\rangle\}$. The mocap-weight pair $\langle CF_i, W_i\rangle$ describes the mapping between the PCA coefficients of one motion capture frame, CF_i, and its blendshape weights, W_i. Given a reference face image V_i, we manually tuned blendshape weights W_i to perceptually match the animated faces and the reference face images. Figure 5 illustrates some examples of tuned blendshape faces, the reference video frames, and the reference motion capture frames.

After the motions of all markers in one frame are packed into a motion vector, Principal Component Analysis (PCA) is applied onto all

the motion vectors to reduce its dimensionality. We experimentally set the reduced dimensionality to four, which covers 90.21% of the variation. Figure 3 plots the reduced motion vectors in the eigen-space spanned by the largest three eigen-vectors. Therefore, given the calculated mean vector $MEAN$ and the reduced eigen-vector matrix $EigMx$, we projected the selected reference motion capture frames $\{C_i\}$ to the above four dimensional eigen-space (Eq. 3) to get reduced vectors $\{CF_i\}$. Here CF_i is a four dimensional vector (projected PCA coefficients). Finally, we obtained 100 correspondence pairs between CF_i (4 dimensional PCA coefficients) and W_i (46 dimensional blendshape-weight vectors). We referred to these pairs $\{\langle CF_i, W_i\rangle\}$ as *reference mocap-weight pairs*, used in the follow-up section as training sets.

$$CF_i = EigMx.(C_i - MEAN) \tag{3}$$

It is a difficult task to determine proper reduced dimensionality. Because retaining higher dimensionality keeps more subtle motion in the reduced vectors and covers more percentage of the motion variation. However, more training sets are needed to train RBF regression functions in Section 6, since the input vector space is enlarged accordingly. It is a trade-off between the reduced dimensionality of motion capture data and the number of training sets. Based on the chosen 100 training pairs, we tried several different reduced dimensionality in order to experimentally determine proper reduced dimensionality, and found that experimental results from lower dimensionalities (e.g. 2 and 3) significantly dropped expression details, and those of higher dimensionalities occasionally introduced unsatisfactory combinations at some frames. As a trade-off, we experimentally set it to 4.

6 Radial Basis Function Regression

In this section, we describe the calculation of blendshape weights for an input motion capture frame using scattered data interpolation algorithms [Nielson 1993]. The reference mocap-weight pairs $\{\langle CF_i, W_i\rangle\}$ are used as training sets.

There are a number of options for this interpolant, such as B-splines [Lee et al. 1997], Shepard's method [Beier and Neely 1992], and Radial Basis Functions [Buhmann 2003]. Since Radial Basis Functions (RBF) have been shown to have successful applications in facial animation [Pighin et al. 1998; Noh and Neumann 2001; Lewis et al. 2000], we decided to use a RBF regression technique to calculate blendshape weights for input motion capture frames.

The network of RBFs can be described as follows (Eq. 4):

$$F(X) = \sum_{k=1}^{N} w_k \phi_k(X) \tag{4}$$

Here we use the Gaussian function as the basis function, $\phi_k(X) = \exp(-\|X - CF_k\|^2 / 2\sigma_k^2)$, N is the number of the training sets, w_k is the weight to be calculated, and $F(X)$ is the estimated function output. The above regression function output $F(X)$ (Eq. 4) is created for each blendshape weight.

Ridge regression (weight decay) and the regularisation parameter λ are used to control the balance between fitting the data and avoiding the penalty. Taking the regression function, $F_j(X)$, for the j^{th} blendshape weight as an example, we want to minimize the following cost function (Eq. 5):

$$C = \sum_{k=1}^{N} (W_k^j - F_j(CF_k))^2 + \lambda \sum_{k=1}^{N} w_k^2 \tag{5}$$

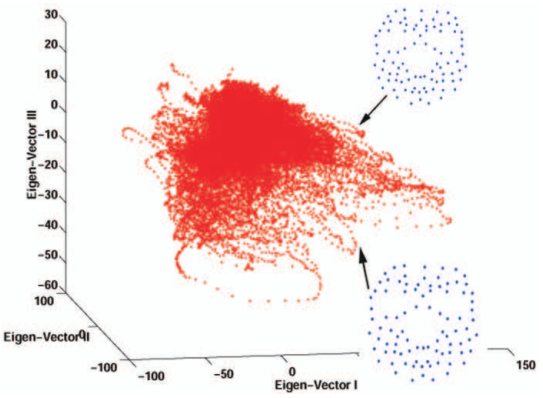

Figure 3: Illustration of reduced motion vectors in the eigen-space spanned by the largest three eigen-vectors. Here each red point represents a facial marker configuration (blue points).

The solution to this minimization problem is Eq. 6.

$$\widehat{w} = (H^T H + \lambda I)^{-1} H^T W^j \tag{6}$$

where $W^j = (W_1^j, \ldots, W_N^j)^T$, and $H_{ij} = \phi_j(CF_i)$. Furthermore, setting the derivative of the Generalized Cross-Validation (GCV) [Buhmann 2003] to zero iteratively re-estimates the regularisation parameter λ until the difference between the current GCV and the previous one is less than a specified small threshold.

In summary, the overall algorithm of mapping any motion capture frame (vector) MC_i to its blendshape weights BW_i can be described in Alg. 1. Here assume there are total M blendshapes for the used blendshape face model.

Algorithm 1 MappingMocapToBlendshapes

Input: a motion capture frame (vector), MC_i
Output: blendshape weights, BW_i
1: Construct N reference mocap-weight pairs, $\{\langle C_i, W_i\rangle\}$.
2: Based on $\{\langle C_i, W_i\rangle\}$, train RBF $f_j(X)(1 \le j \le M)$ (Eq. 4)
3: $X_i = PCA(MC_i)(Eq.\ 3)$
4: **for** j= 1 to M **do**
5: $\quad BW_i^j = F_j(X_i)$
6: **end for**

7 Results and Evaluations

The target blendshape face that we used to evaluate this approach is a NURBS face model composed of 46 blendshapes, such as $\{leftcheekRaise, jawOpen, \cdots\}$. The weight range of each blendshape is $[0, 1]$. Figure 4 illustrates the blendshape face model used in this work. As we can see from Fig. 4, the geometric proportions of the blendshape face model are quite different from that of the captured subject (Fig. 2).

Given a facial motion capture sequence, we first used the approach proposed in this paper to map motion capture frames to blendshape weights. Since the blendshape weights of each frame are solved individually, the smoothness of the weights across frames cannot be guaranteed. Hence, in the postprocessing stage, we use a motion filter algorithm presented in [Bruderlin and Williams 1995] to smooth the blendshape weights across frames.

Besides mapping new motion capture sequences to the blendshape face, we also compared our approach with a widely-used

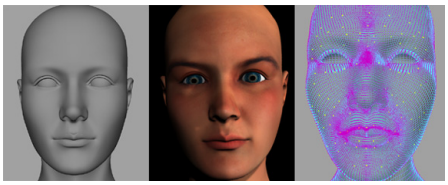

Figure 4: Illustration of the target blendshape face model. The left panel (the smooth shaded model), the middle panel (the rendered model), and the right panel (manually picked "virtual marker" - yellow points - on the polygonized blendshape face model).

blendshape-weight solving method (referred to as *the equation-solving method* in this work) that calculates blendshape weights by solving the standard blendshape equation (Eq. 7). Mathematically, a blendshape model B is the weighted sum of some pre-designed shape primitives B_i:

$$B = B_0 + \sum_{i=1}^{N} w_i * B_i \tag{7}$$

where B_0 is a base face, B_i are delta blendshape bases, and w_i are blendshape weights. Given the real facial marker configuration (the top-right panel of Fig. 2), we manually picked the corresponding vertices in the face model as "virtual markers" (the right panel of Fig. 4). Thus, in Eq. 7, B and $\{B_i\}$ are concatenated vector forms for these virtual markers. B_i is the motion vector when only the i^{th} blendshape weight is set to one (others are zero). Then, the geometric proportions of the blendshape face model to the motion capture subject were measured in X, Y, and Z directions, by calculating the ratio of distances between facial landmark markers, e.g. eye corner markers. After that, motion capture data were scaled using these proportions and packed into B. Finally, we solved the linear equations (Eq. 7) in the least-square sense to get the blending weights $\{w_i\}$.

We compared the mapping results of our approach with the results of the equation-solving method in Fig. 6. As can be seen from the Fig. 6, the face animated by our approach are closer to the recorded faces (ground-truth), especially in the preservation of expression details and characteristics of mouth shapes. More mapping results are illustrated in Fig. 7. For animation results, please refer to the accompanying demo video. Our approach can compute desirable blendshape weights for facial motion capture inputs, but there exist some cases that computed weights by our approach are not perfect, for example, in the fifth image (counting from the top) of Fig. 7, the openness of the mouth is not large enough. However, in some cases, when markers are not perfectly tracked, our approach can automatically deal with it reasonably well, because the regression inputs are in the reduced PCA space, not the original marker motion space. An example is the sixth image of Fig. 7 where red box illustrates the mistracked place of two markers.

8 Conclusions and Discussions

In this paper, we present a semi-automatic technique for directly cross-mapping facial motion capture data to pre-designed blendshape face models. Unlike previous blendshape animation work, this approach is not limited to specific types of blendshape faces. The face models need not be based on muscle actuation bases and the geometric proportions of the blendshape face models may vary from that of the captured subject. We also show that our approach can calculate more proper blendshape weights than a widely-used weight-solving method.

The reference mocap-video pairs and mocap-weight pairs introduced in our work are used to bridge facial motion capture data and blendshape face controls. Manual work is required in this process. However, the cost of the manual work is affordable, considering large motion capture databases often consist of hundreds of thousands of motion capture frames. Instead manually tuning weights every two or three frames, this approach can be used as a fast animation tool given facial motion capture sequences. Another advantage of this approach is that it automatically maintains the coherence of blending weights across frames, while this coherence could cause a problem for animators' manual-tuning work.

Currently this approach works on 3D facial motion capture data, but it could be extended to enhance performace-driven facial animation [Williams 1990], by directly cross-mapping tracked facial motion from video to blendshape models. Probably, the most difficult challenge would be the normalization and alignment of the tracked facial motion in order to force all motion data into the same coordinate system.

The retained dimensionality of the principal component analysis used in constructing reference mocap-weight pairs (Section 5) is important to the efficiency of this approach. As discussed in Section 5, it is a difficult trade-off between the reduced dimensionality of motion capture data and the number of training sets. As a future work, we plan to investigate new ways to optimize the reduced dimensionality for this cross-mapping, e.g., cross-validation.

The mapping from the facial motion capture data space to the blendshape weight space is generally a one-to-many mapping. The individual preferences of animators in choosing blendshape controls is a factor when they are constructing the reference mocap-weight pairs that are later used to train RBF regression functions. Hence, animators with different blendshape-control preferences may get different cross-mapping results given the same motion capture data input. Mixing reference mocap-weight pairs constructed by different animators together involves a risk of causing conflicts in the RBF regression training and may result in unpleasant cross-mapping results.

There remains a number of interesting extensions to this work that can be pursued in the future. Current approach still requires manual work in the setup stage, including selecting reference motion capture frames and tuning blendshape weights. We plan to work on automatically extracting the best reference motion capture frames from a motion capture database. Additionally, in our current work, we also assume that the scheme of selecting reference motion capture frames can achieve its goal - covering the spanned space of facial movements as balanced and completely as possible. Future work can develop an evaluation tool/algorithm to quantatively measure the coverage of reference mocap frames and iteratively refine the selection scheme. A blendshape face model composed of 46 blendshapes was used in this work, while blendshape models used in current industry practice are more complex, often composed of hundreds of blendshapes. Hence, future work can be done to evaluate and improve this approach on these various complex blendshape face models.

Acknowledgements

This research has been funded by the Integrated Media System Center at USC, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Special Thanks go to J.P. Lewis for thoughtful discussions, Joy Nash, Murtaza Bulut, and Carlos Busso for facial motion data capture and processing, Hiroki Itokazu, Bret St. Clair, and Shawn Drost for face model preparation.

References

BEIER, T., AND NEELY, S. 1992. Feature-based image metamorphosis. In *Computer Graphics (ACM SIGGRAPH '92)*, ACM Press, 35–42.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proc. of ACM SIGGRAPH'99*, 187–194.

BLANZ, V., BASSO, C., POGGIO, T., AND VETTER, T. 2003. Reanimating faces in images and video. *Computer Graphics Forum(Proc. of Eurographics 2003) 22*, 3.

BRAND, M. 1999. Voice puppetry. In *Proc. of ACM SIGGRAPH 1999*, 21–28.

BREGLER, C., COVELL, M., AND SLANEY, M. 1997. Video rewrite: Driving visual speech with audio. In *Proc. of ACM SIGGRAPH'97*, 353–360.

BRUDERLIN, A., AND WILLIAMS, L. 1995. Motion signal processing. In *Computer Graphics (ACM SIGGRAPH '95)*, ACM Press, 97–104.

BUHMANN, M. 2003. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press.

BUSSO, C., DENG, Z., NEUMANN, U., AND NARAYANAN, S. S. 2005. Natural head motion synthesis driven by acoustic prosodic features. *Computer Animation and Virtual Worlds (special issue for best papers of CASA 2005) 16*, 3-4 (July), 283–290.

CAO, Y., FALOUTSOS, P., KOHLER, E., AND PIGHIN, F. 2004. Real-time speech motion synthesis from recorded motions. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 345–353.

CHAI, J., XIAO, J., AND HODGINS, J. 2003. Vision-based control of 3d facial animation. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 193–206.

CHOE, B. W., AND KO, H. S. 2001. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *IEEE Computer Animation Conference*, 12–19.

DECARLO, D., AND METAXAS, D. 1996. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 231–238.

DENG, Z., LEWIS, J. P., AND NEUMANN, U. 2005. Synthesizing speech animation by learning compact speech co-articulation models. In *Proc. of Computer Graphics International (CGI 2005)*, IEEE Computer Society Press, 19–25.

ESSA, I., BASU, S., DARRELL, T., AND PENTLAND, A. 1996. Modeling, tracking and interactive animation of faces and heads: Using input from video. In *Proc. of IEEE Computer Animation'96*, 68–79.

EZZAT, T., GEIGER, G., AND POGGIO, T. 2002. Trainable videorealistic speech animation. *ACM Trans. Graph.(Proc. of ACM SIGGRAPH'02) 21*, 3, 388–398.

FESTIVAL. http://www.cstr.ed.ac.uk/projects/festival/.

KSHIRSAGAR, S., AND THALMANN, N. M. 2003. Visyllable based speech animation. *Computer Graphics Forum (Proc. of Eurographics'03) 22*, 3.

LEE, S., WOLBERG, G., AND SHIN, S. Y. 1997. Scattered data interpolation with multilevel b-splines. *IEEE Transaction on Visualization and Computer Graphics 3*, 3, 228–244.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of ACM SIGGRAPH'2000*.

LEWIS, J. P., MOOSER, J., DENG, Z., AND NEUMANN, U. 2005. Reducing blendshape interference by selected motion attenuation. In *Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, 25–29.

LI, H., ROIVAINEN, P., AND FORCHHEIMER, R. 1993. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 15*, 6, 545–555.

MA, J., COLE, R., PELLOM, B., WARD, W., AND WISE, B. 2004. Accurate automatic visible speech synthesis of arbitrary 3d model based on concatenation of diviseme motion capture data. *Computer Animation and Virtual Worlds 15*, 1–17.

NIELSON, G. M. 1993. Scattered data modeling. *IEEE Computer Graphics and Applications 37*, 4, 60–70.

NOH, J. Y., AND NEUMANN, U. 2001. Expression cloning. In *Proc. of ACM SIGGRAPH'01*, 277–288.

NOH, J. Y., FIDALEO, D., AND NEUMANN, U. 2002. Gesture driven facial animation. *USC CS Technical Report 02-761*.

PARKE, F. I., AND WATERS, K. 1996. *Computer Facial Animation*. A K Peters, Wellesley, Massachusets.

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

PIGHIN, F., SZELISKI, R., AND SALESIN, D. 1999. Resynthesizing facial animation through 3d model-based tracking. In *IEEE International Conference on Computer Vision (ICCV)*, 143–150.

PYUN, H., KIM, Y., CHAE, W., KANG, H. W., AND SHIN, S. Y. 2003. An example-based approach for facial expression cloning. In *Proc. the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, Eurographics Association, 167–176.

SCOTT, R. 2003. Sparking life: notes on the performance capture sessions for the lord of the rings: the two towers. *ACM SIGGRAPH Computer Graphics 37*, 4, 17–21.

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

STEGMANN, M. B., AND GOMEZ, D. D. 2002. A brief introduction to statistical shape analysis. In *Informatics and Mathematical Modelling, Technical University of Denmark, DTU*.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3, 399–405.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Transaction on Graphics 24*, 3.

WILLIAMS, L. 1990. Performance-driven facial animation. In *SIGGRAPH '90*, ACM Press, 235–242.

Figure 5: Three examples of tuned blendshape faces (the top), the reference face images (the middle), and reference motion capture frames (the bottom).

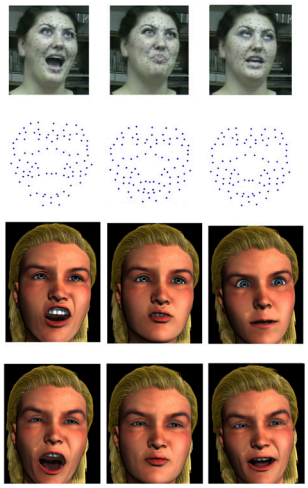

Figure 6: Comparisons of grouthtruth video, motion capture data, results of the equation-solving method, and results of our approach. The first row shows recorded groundtruth video frames, the second row shows aligned motion capture frames (for comparison purpose, head motion was not removed in these frames), the third row shows the mapping results of the equation-solving method, and the fourth row shows the mapping results of our approach.

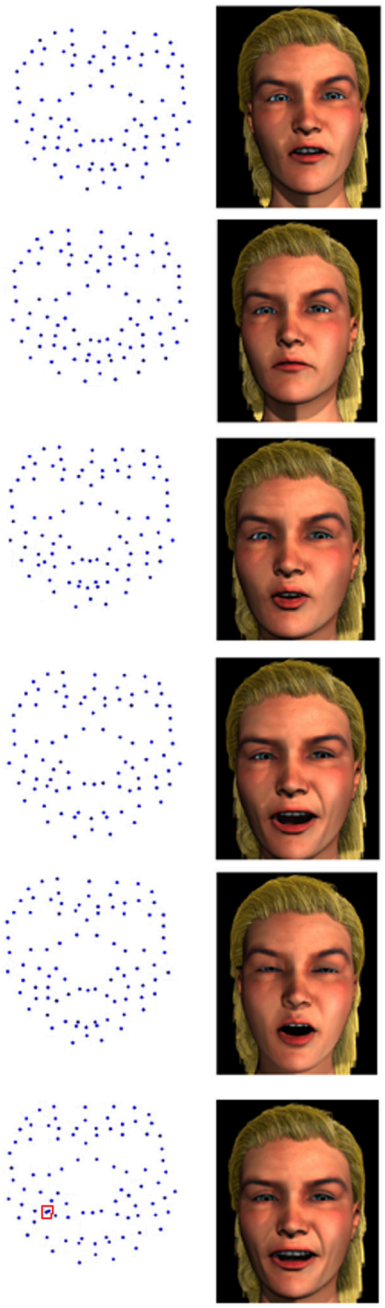

Figure 7: Examples of facial motion capture mapping results. The left column shows motion capture frames, and the right column shows corresponding mapping results generated by our approach.