Lips-Sync 3D Speech Animation using Compact Key-Shapes

Fu-Chun Huang* Bing-Yu Chen* Yung-Yu Chuang*
National Taiwan University

Abstract

Facial animation is traditionally considered as important but tedious work for most applications, because the muscles on face are complex and dynamically interacting. Although there are several methods proposed to ease the burden from artists to create animating faces, non of these are fast and efficient in storage.

This paper introduces a framework for synthesizing lips-sync facial animation given a speech sequence. Starting from tracking features on training videos, the method first find representative key-shapes that is important for both image reconstruction and guiding the artists to create corresponding 3D models. The training video is then parameterized to weighting space, or cross-mapping, then the dynamic of features on the face is learned for each kind of phoneme.

The propose system can synthesis lips-sync 3D facial animation in very short time, and requires very small amount storage to keep information of the key-shape models and phoneme dynamics.

Keywords: lips-sync, speech, animation, 3D

1 Introduction

With the popularity of 3D animation movie and video games, facial animation is becoming more important than ever, in order to lively animate the virtual character. It is, however, a laborious task to create a 3D face model, let alone a sequence of models as an animation. Facial animation is difficult to sculpt, because the correlation and complexity of the muscle on the face is so high that few mathematical model can approximate it without extensive computation. Although there exists a wide range of work based on *physical simulation*, the computation cost is still exorbitant to general users.

A less flexible but affordable alternative is *performance driven* facial animation that the motion of the actors are somehow transferred to the virtual character, such as *The Polar Express* and *Monster House*. Although these animation have successfully convinced our imagination, they are not reusable and new performance is required each time at creating novel sequence.

In this paper a method that build a re-usable model for animating faces with speech is proposed. While traditional performance-driven facial animation can only drive the virtual face to deform accordingly, the proposed method, starting with performance to train the speech deformation space, can synthesize any new animation given novel speech sequence. The method not only synthesize animation respecting to the visual-phoneme correspondence, but it also refine the co-articulation effect that is poorly modeled in similar previous work, and the highlight is *lip-sync speech animation.*

The input of the proposed method requires a video sequence, some feature point positions in the video sequence, and a mere number of key-face model that is made respecting to some key-shapes in the video sequence found by the method. An animated speech animation given novel speech sequence is generated as output.

The major contributions includes:

1. A method for key-shape identification that traditionally can not be achieve without much parameter tuning and trial-error process is proposed. Our method is fast and require only one pass.
2. A cross-mapping that transfer animation from one subject of the performer to the other, such as virtual character, is illustrated. While most previous work primarily focus on weight-space tuning using non-linear function, such as Radial Basis Function, we propose the use of exponential-map that have physical meaning.
3. An attempt to animate 3D models with training data merely from 2D videos is made, while previous work making efforts on image space lips-sync video speech animation.

2 Related work

The book *MPEG-4 Facial Animation: The Standard, Implementation and Applications* [Pandzic and Forchheimer 2002] provides a clear categorization of fields dealing with facial animation (FA). High-level FA is separated from Low-level FA through the use of control parameterizations, and the following subsections will briefly describe several techniques in each fields, as in Fig. 1

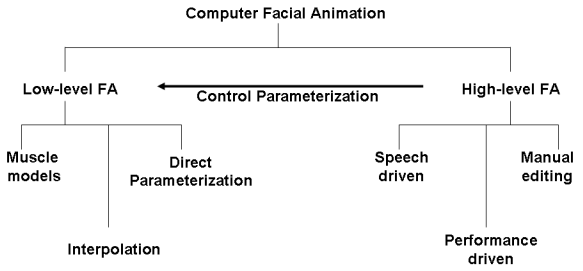

Figure 1: This figure briefly summarizes Low-level FA and High-level FA, while most research combine methods from two categories.

2.1 Low-level FA

Low-level FA deals with methods to change the facial geometry and/or appearance in time and the control relies on the extrinsic parameters. Important approaches include Direct Parameterizations, Pseudo-Muscle/Muscle Model, and Interpolation. Direct parameterizations usually refers to the creation of facial model based on the intuition of the artists, and mostly appeared as spline surfaces widely used in animation industry. While the most time-consuming, direct parameterizations also is the most popular technique.

In recent year the Interpolation approach starts to gain its popularity due to its simplicity and also the possibility provided to use highly detailed models, yet there are still areas that cannot be modeled without the Muscle Model.

*e-mail: {jonash,robin,cyy}@cmlab.csie.ntu.edu.tw

2.1.1 Muscle based approaches

Muscle based approaches strive to mimic the minutia on face through bio-chemical and anatomic details. Choe et al.[Choe et al. 2001] uses two kinds of muscle with finite element method to learn the captured motion signal. A significant effort is devoted to establish the relationship between muscle actuation and surface deformation. Their heuristic approach, however, lacks anatomical structure and need additional effort to adjust correlation near the mouth. More recently, Sifakis et al.[Sifakis et al. 2005] propose an anatomical accurate approach to analyze the signals of muscle activation which correspond to the sparse landmark on the face. Although their result is particularly useful in physical simulation, as illustrated in the paper from Sifakis et al.[Eftychios Sifakis and Fedkiw 2006], the speed of this sort is prohibitive unaffordable when time is a major concern.

2.1.2 Interpolation based approaches

Interpolation based approach poses another advantage over that of muscle based: simplicity and efficiency. Even the amount of blend-shape might be huge (more than hundreds of shape were used in *The Lord of the Rings* for the *Gollum* creature), the Interpolation based community still gains its popularity. The Facial Action Coding System (FACS) by Ekman and Friesen[Ekman and Friesen 1977] provide a guideline for sculpting the blend-shape, where 72 Action Units of distinct expressive faces are categorized, and this technique is also used in the film industry, for example *Monster House*. Among the several methods using Interpolation approach, there can still be a fundamental distinction between image-based and model-based.

Image-based FA has the advantage in data acquisition as opposed to that of 3D-Model based, and the algorithm work efficiently when projected to the PCA space. Ezzat et al.[Ezzat et al. 2002] show how to reanimate a person by decomposing video sequence into key-shape space, and propose a Gaussian-Phoneme Model to synthesize new speech animation using Multidimensional Morphable Models(MMM) by Jones and Poggio[Jones and Poggio 1998]. Their work is extended by Chang and Ezzat[Chang and Ezzat 2005] to transfer model to different faces with existing trained model. Buck et al. [Buck et al. 2000] also shows how to blend sample shapes for hand-drawn animation with parameterizations using Delaunay triangulation in the feature space.

Model-based FA include work by Chai et al.[xiang Chai et al. 2003], Guenter et al.[Guenter et al. 1998], and Pighin et al.[Pighin et al. 1998], and etc,. PCA based method by Blanz and Vetter[Blanz and Vetter 1999] extends MMM[Jones and Poggio 1998] to 3D model using morphable model database with feature-controls, and their method is able to reconstruct and animate photograph. Zhang et al.[Zhang et al. 2004] propose a FaceIK framework with blend-shape scanned from their space-time stereo approach. They also introduce adaptive face segmentation to reduce interference from unimportant blend-shapes. In order to reduce the blend-shape interference, Lewis et al.[Lewis et al. 2005] minimize the error incurred by selected control vertices with user controllable scaling factor.

2.2 High-level FA

High-level FA controls the parameters derived from Low-level FA modeler to direct the face motion. The most precise but also the most time-consuming is manual editing, with artists introduce numerous key-frames to capture the non-linearity on facial motion. To reduce cost incurred from labor, two mainstreams are Performance-driven and Speech-driven approaches.

2.2.1 Speech-driven

Speech-driven approach has its advocates because the fact that they are mostly generative model, i.e. animators can generate new FA without tedious parameterizations from motion capture.

Papers by Bregler et al.[Bregler et al. 1997] and Brand[Brand 1999] are two important precursors in this field. Ezzat et al.[Ezzat et al. 2002] build a Gaussian based phoneme model and synthesize animation given phoneme aligned speech. The result of previously built model can be transferred to a novel person with a short training clip and synchronized speech as described in the paper by Chang and Ezzat[Chang and Ezzat 2005]. The original model parameters are adjusted in accordance with the novel video corpus.

Cao et al.[Cao et al. 2004] propose a fast searching algorithm in their motion database for synthesizing new animating facial sequence. The eFASE system by Deng and Neumann[Deng and Neumann 2006] also synthesizes new faces from recorded motion in database, and a path searching algorithm in the phoneme space using Isomap projection is proposed.

2.2.2 Performance-driven

Performance-driven approaches usually starts with a motion capture process. The signals of captured data are transferred as controls for reanimating FA with a cross-mapping function. Finally the cross-mapped signals are sent to Low-level FA modeler as rigs to modify geometry data. Among the many performance driven approaches, the issue of cross-mapping is usually at heart to solve.

Williams *"Performance-Driven Facial Animation"*[Williams 1990] is the very first research in this field. In the course note[Pighin and Lewis 2006], several survey of techniques for cross-mapping function are presented. The simplest function form is linear as in the report by Chuang and Bregler[Chuang and Bregler 2002]. Vlasic et al. [Vlasic et al. 2005] use multi-linear, or tensor, model to construct a statistical model for human faces, the dimension can be identity, expression, viseme, and etc,. The cross-mapping is done with motion capture from one identity and transfer to another within framework.

Choe et al.[Choe et al. 2001] build the inverse relationship with radial basis function (RBF), and also Pyun et al.[Pyun et al. 2003], Na and Jung[Na and Jung 2004], and Deng et al.[Deng et al. 2006], by estimating muscle actuation profile from the performance, so that performance signal can directly be translated as the simulation parameters.

An interesting technique proposed by Buck et al.[Buck et al. 2000] to transfer expression from performer to an NPR character. They project the expression space on to the first two principal component, where Delaunay triangulation is subsequently applied. Whenever a new novel face is to be estimated, barycentric coordinate to its nearest three key-shapes is calculated. This coordinate is afterward translated to blending weights.

2.2.3 Geometry Transfer

Expression Cloning by Noh and Neumann[Noh and Neumann 2001] and Deformation Transfer for Triangle Meshes by Sumner

and Popović[Sumner and Popović 2004] are two papers that use direct geometry transfer function rather than parameterizing the transfer function. Noh and Neumann[Noh and Neumann 2001] estimate the local geometry property and transfers the motion vector to the target mesh with constraints that respect the locally defined boxes in the two models. Sumner and Popović[Sumner and Popović 2004] extract the local gradient from the source and estimate similar gradient on the target mesh. Local misalignment of the triangle gradient is solved using global optimization framework with vertex connectivity as constraint, small imperfection on the mesh can, however, amplified, as noted in the course note[Pighin and Lewis 2006].

2.3 Summary

In this paper, a framework incorporating Interpolation and Speech-Driven animation is presented. The technique used in Interpolation will be presented in Section 3, where a brief discussion on linear/non-linear function will be given. As for high-level facial animation, the statistical analysis for speech and its corresponding key-frames will be studied in Section 5. Before the analysis, some preprocessing issue will be discussed in Section 4.

3 Preliminary

3.1 Animation Reconstruction from Scattered Data Observation

One key process, to drive the 3D model animate, is to control the 3D model with certain given constraints. In light of cost/convinences, constraints may be sparse but should be representative enough for surface reconstruction. One popular solution is Radial Basis Function(RBF), where intended values are interpolated from kernel-convoluted distances to scattered observation. The problem of animation reconstruction could, however, be relaxed in a way when we already know some configuration of the surface. PCA based techniques have been proposed to drive 3D model, yet numerous aligned examples have to trained in advance. (try to cite SCA 2006 by UW, and SCAPE) Example-based model interpolation provide an alternative without extensive preprocessing. Throughout this paper, we will always focus on processing triangle mesh.

An explicit way to represent triangle mesh is with the coordinates of its vertices in the global frame, and to reconstruct model surface, we can transform the problem into minimization, given K example meshes $\{P_1,...,P_K\}$:

$$\alpha = argmin_{\alpha_1^*,...,\alpha_k^*} \|C - \sum_{i=1}^{K} X_i \alpha_i\|$$

where C is the constrained vertices position, $X_i \in P_i$ is the constrained vertices position in example mesh, and α is the intended weights to blend these example meshes as reconstructed mesh. Problem with such approach is its lack of capturing local shape property and relation, and when interpolating extreme poses, discontinuity in surface will occur due to the global frame position misalignment, and sample result is shown in Fig. 2 middle. While linear-interpolation works well with ***very dense*** examples, it usually increase the complexity and cost to acquire these models.

To describe mesh better, it is suitable to represent mesh as a vector in another feature space based oh their local shape property. We use deformation gradient, as the paper by Barr[Barr 1984] but in a discrete form, to represent triangle mesh.

Figure 2: Left: Original model to be approximated. Middle: Interpolated model using global frame without local-shape-preservation. Right: Non-linear interpolated model solved with Eq. 2

3.2 Deformation Gradient

Given a mesh in reference pose P_0 and a deformed mesh P, both having n vertices and m triangles in identical connectivity structure, a deformation gradient is the Jacobian of the affine transformation, from a triangle in P_0 to P.

Denote the affine transformation Φ^j of the triangle τ^j to map points it is defined $v_k^j \in \tau^j, k = 1,2,3$ as:

$$\Phi^j(v_k^j) = T^j v_k^j + t^j$$

T^j is the rotation/scale/skew component and t^j is the translation. Taking Jacobian of Φ^j with respect to v^j results in T^j alone. the feature vector F_i of the mesh P_i is constructed by concatenating Jacobian of each triangles in a column form. Getting position back from feature vector consisting only Jacobian is done through integration, as described by Barr[Barr 1984], or by Sumner et al.[Sumner and Popović 2004] for discrete form. In computation, the discrete form is as describe as:

$$V = argmin_{v^*} \|Gv - F\|$$

where G is built from the reference pose P_0 and v is the vertices postilion we want to recover from feature vector F.

Using deformation gradient, F_i for example P_i, as mesh descriptor for animation reconstruction given constraints, we can formulate the problem mathematically as :

$$\alpha = argmin_{\alpha_1^*,...,\alpha_k^*} \|Gv - \sum_{i=1}^{K} F_i * \alpha_i\|$$

and the minimization is subject to $X \in v$ be equal to constraints C. Although this description of mesh can preserve local shape property, artifacts can still appear at interpolating two surfaces with large deviation in orientation where no intermediate examples are available.

Non-linear interpolation, specifically exponential-map or log-matrix blending used in this paper, is designed to solve the problem. Matrix representing the Jacobian of affine transform can be factored into rotation and skew components through *polar decomposition* by Shoemake and Duff[Shoemake and Duff 1992], thus the Jacobian of affine transform for triangle τ^j in mesh P_i is separate as $T^{ij} = R^{ij}S^{ij}$. The rotation component R^{ij} is blended in log-matrix

space whereas the skew component S^{ij} in Euclidean space. The non-linear convolution is computed as follows:

$$M(T^j,\alpha) = exp(\sum_{i=1}^{K} log(R^{ij})\alpha_i)(\sum_{i=1}^{K} S^{ij}\alpha_i)$$

Using exponential map to blend example meshes, we can get our result mesh as :

$$V,\alpha = argmin_{v^*,\alpha^*}\|Gv - M(F,\alpha)\| \quad (1)$$

subject to $X \in v$ be equal to constraints C. Details and solution to the non-linear minimization is refereed to Sumner et al.[Sumner et al. 2005], where a Gauss-Newton method was utilized to linearizes the equation and solved in a iterative manner. In latter discussion, result α will be used as parameter for our training. Given α, we are able to recover position v by first computing the deformation gradient $F = M(F,\alpha)$, and plugging in Eq.2 to solve directly without iteration.

$$V = argmin_{v^*}\|Gv - M(F,\alpha)\| \quad (2)$$

4 Capture and Preprocessing

A SONY DCR TRV 900 video camera is used for the capture purpose. The camera is placed in frontal direction toward face of the tracked subject. The subject is placed with 15 landmarks around the lips and jaw for easier tracking, as in Fig 3, though not absolutely necessary. The result contains about 10 minutes video, approximately 18000 frames. The subject is asked to speak contents that elicit no emotion, and these contents also includes bi-phone and tri-phone to reflect co-articulation.

4.1 Feature Tracking

The result of recording video is subsequently tracked using a customized tracker. Although various tracking algorithm exist, Lucas Kanade Tomasi(KLT) tracker, a direct tracking method based on constancy of feature brightness, is used in the paper.

Other popular tracker such as Active Appearance Model(AAM)/variation, a trainable model that separates structure of images from texture and learns basis using PCA, is ideal for on-line purpose, is, however, not necessary for our database building.

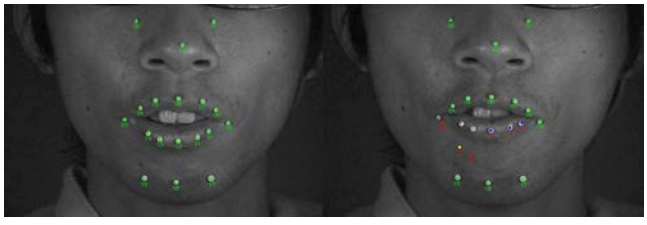

Figure 3: 18 landmarks are placed on the face, 3 for position stabilization, and 15 are used for tracking (12 around lips, and 3 on jaw). Sometimes mis-tracked/lost features require the user to bring back.

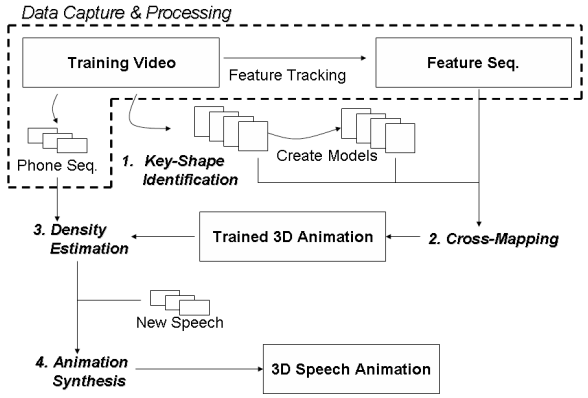

Figure 4: The figure indicate the overview of our system including the data capture and processing step.

4.2 Phoneme Segmentation

Each frame of the recorded sequence corresponds to a phoneme the subject speaks, and given transcript, the CMU SphinxII can decode the audio and cut the utterance into a sequence of phoneme segments. The CMU SphonxII use 51 phonemes, including 7 sounds, 1 silence, and 43 phonemes, to construct the dictionary.

The result of preprocessing after feature tracking and phoneme segmentation contain a list of data of 15 dimensionality for each frame associated with a specific phoneme Sphinx II defines.

5 Algorithm

5.1 Overview

After the capture of the performer and subsequent tracking and phone alignment, the artist is ready to create certain 3D face models that can be useful for interpolating examples. In section 5.2 a discussion of how to cluster training data without knowing how many groups needed is given, where affinity propagation [Frey and Dueck 2007] is exploited. The result instructs artists to sculpt 3D face models that correspond to certain key-shapes in the training sample, and a method in section 5.3 is introduced to solve the following analogy:

$$Training_{exp} :: Training_{neut} = \mathbf{?} :: FaceModel_{neut}$$

Such analogy is solved with the algorithm from Sumner et al.[Sumner et al. 2005], and we can obtain a high dimensional parameter for each frame of the training data that correspond to each phoneme the performer speak. After that, relationship can be built with Gaussian estimation for each phoneme, similar to the work by Ezzat et al.[Ezzat et al. 2002]. In section 5.4, an energy function is presented to synthesize any novel speech for new parameters that can composite a sequence of 3D facial animation.

5.2 Prototype Image-Model Pairs Identification

Since our method highlights **example-based interpolation**, representative prototype lips-shapes as key-shapes have to be identified. Without lose of generality, fewer key-shapes are desired, since to

sculpt 3D models as key-shape is painstaking; more key-shapes are needed, to vividly span the spectrum of facial animation. For convenience, the representative key-shapes are selected from the training data set, and later used for any-shape reconstruction.

However, finding representative key-shape is not easy, and traditional K-center algorithm does not work efficiently and requires many runs. Researchers may sometimes reduce finding prototype key-shape S_i into the following minimization problem :

$$\min \sum_j ||f_j - \sum_{i=1}^{K} S_i w_{ij}||^2 \quad (3)$$

In the formulation, f_j corresponds to a frame in our data set, w_{ij} is the weight to linear blend key-shapes $\{S_1, ..., S_K\}$ approximating frame f_j. The formulation, though simple, has one potential problem on one hand that each cluster is not normalized to influence the entire data set minimization; on the other hand, there are no other choices. Besides, the number of key-shapes K is still unknown in advance. In this paper, the following questions need to be answered:

1. How many prototype key-shape are needed?
2. What are these key-shapes and the clusters key-shapes define?
3. Which data point belongs to which cluster?

Ezzat et al.[Ezzat et al. 2002] heuristically determine K with prior experimental experience, cluster data set using k-means algorithm, and assign prototype key-shapes to data points *nearest* to cluster centers.

Chuang and Bregler[Chuang and Bregler 2002] proposes the data with *Maximum spread along principle component* is preferred as key-shape, although K is still determined heuristically. Others like *Convex-hull*, though perform no better than Max. spread along PCA, still indicates a viable alternative.

In solving the three problem simultaneously, *Affinity Propagation* [Frey and Dueck 2007], is utilized to help the identification of K, these key-shapes, and clusters.

Affinity Propagation

Affinity propagation is a bottom-up algorithm iteratively merges points/sub-clusters into clusters. In between data point, two sorts of messages are passed, *Responsibility* and *Availability*.

Responsibility $r(j,i)$ sent from data point j to exemplar, or key-shape in our case, i are messages reflects the accumulated evidence for how well-suited point i is to serve as the exemplar for point j, taking into account other potential exemplars for point j.

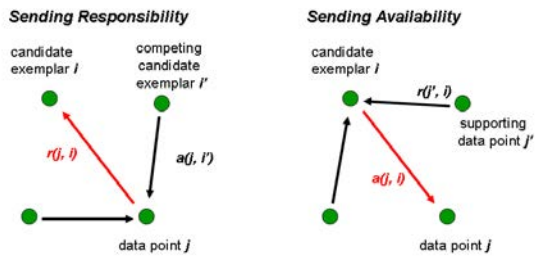

Figure 5: This figure illustrates two kinds of messages are sent to each data point.

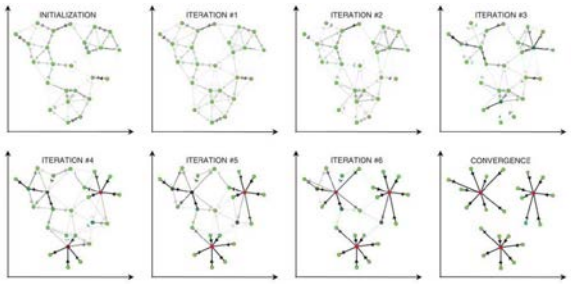

Figure 6: This figure briefly show how data set are clustered through affinity propagation.

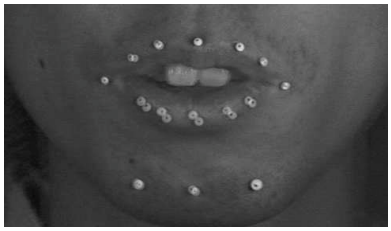

Figure 8: Affinity propagation surprisingly finds shapes that are very close but indeed different. The image is an overlay of two close group.

Availability $a(j,i)$ sent from exemplar i to point j are messages reflect accumulated evidence for how appropriate it would be for point j to choose point i as exemplar, taking into account the support from other points that point i would be exemplar.

The Figure 5 show how messages are sent and and data set is iteratively clustered in Fig.6.

The algorithm is fast and work without much parameters tuning. The only input is a pair-wise similarity table that specifies how each point is similar to others. Self-similarity indicates how much evident a point believes itself as an exemplar, and in this paper the value is set to the minimum of pair-wise similarity, as instructed in Frey and Dueck[Frey and Dueck 2007] for smaller number of clusters.

Affinity propagation successfully separates 18000 tracked speaking lips-shapes into 21 clusters, in Fig. 7, with each defines a group either small or large. It is less likely for other methods to find *small yet representative groups* since most error minimization are done with respect to the entire data set and error are biased to favor large group.

There is also an interesting finding that affinity propagation separates groups with very subtle difference, that when talking the lips shape are not symmetric but biased to either left or right, as Fig.8, due to the unsymmetrical muscle activation on human.

It might be interesting but not practical for artists to sculpt such minutia on 3D models, and further reduction on the number of key-shapes is desired. Merging groups with minimum distance on key-shapes is performed iteratively until either a threshold is reached or a sudden increase in reconstruction error occurs (see Fig.10), and finally 7 key-shapes,as in Fig. 11, are identified. A comparison of affinity propagation with direct minimization on Eq.3 is given in Fig.9. Finally the artists create 3D facial model, as in Fig. 11, with lip-shapes according to the 7 key-shapes identified.

There is another algorithm, namely Mean-Shift Clustering

Figure 7: The total key-shapes found by affinity propagation. As seen, there are many looks very similar yet actually different.

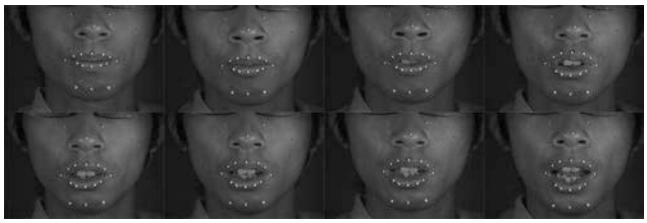

Figure 9: The result of key-shapes from directly minimizing Eq.3, where the almost 8 evenly keyed close to open result was found. Certain important key-shapes like *woo* or *foo* are missing.

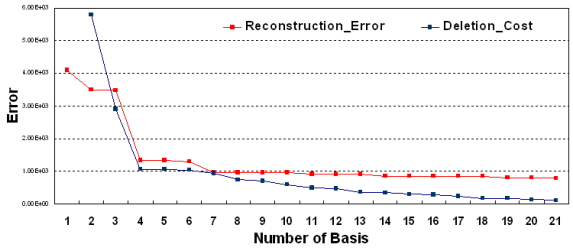

Figure 10: This graph show the reconstruction of reducing the number of redundant basis, as in red line. The cost of removing each basis, by choosing least group distance, is shown in blue. The error of reconstruction keep steady until 7 basis remains. The basis deletion cost climbing up steadily and reach a plateau of high cost around 7.

[Georgescu et al. 2003], worth mentioning to identify these representative groups without knowing K in advance. James and Twiggs[James and Twigg 2005] successfully identifies the bones/joints in articulated animating sequence, and the algorithm is also exploited by Park and Hodgins[Park and Hodgins 2006]. Difficulty with such alternative is the un-intuitive and data dependent parameter *Bandwidth*, but nevertheless we also identify 20 key-shapes in the data set, compared with 21 using affinity propagation though the computation is slower.

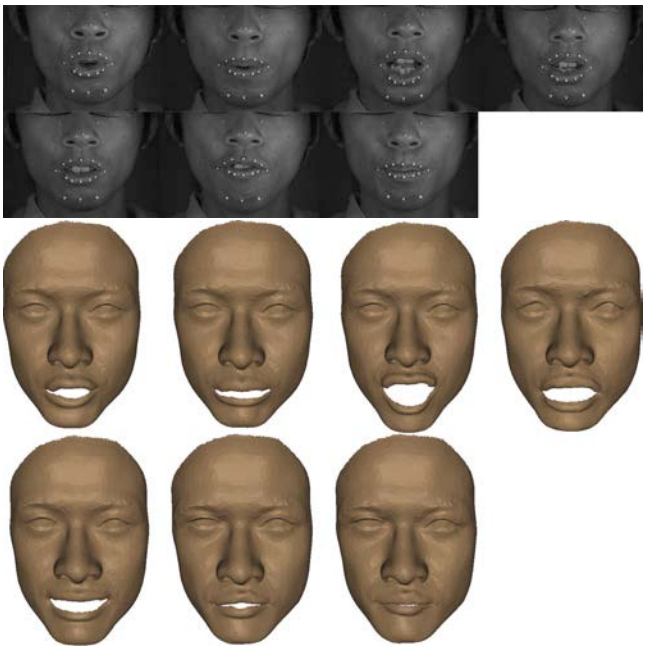

Figure 11: One-to-one mapping of the key-shape from video sequence and the artists created 3D meshes.

5.3 Training Sample Parameterizations and Cross Mapping

After identifying key-shapes, the sequence of 3D animation, given the sculpted 3D model, can be created according to the training video given the key-shape images. The motion of the training video is cross-mapped to 3D model as discussed in this section.

The jargon *Cross Mapping* in facial animation usually refers to transfer the motion of one subject to the other. The typical case is an actor giving a performance and the goal is to use such to animate a virtual character. The user may desire a such function, if any, to transfer even very minute details, and such function is adjustable so that one performer can drive a numerous virtual character without too much complication.

Pyun et al.[Pyun et al. 2003], Na and Jung[Na and Jung 2004], and Lewis et al.[Lewis et al. 2005] use Radial Basis Function for fine-tuning the weight-space in each dimension and require more train-

ing samples to explore the function form. Buck et al.[Buck et al. 2000] use convex hull technique that each training faces is project to a plane, with three nearest key-shapes, the barycentric coordinates are obtainable and transferred to NPR faces for rendering. Chuang and Bregler[Chuang and Bregler 2002] propose a cross mapping algorithm most representing ours, yet the parameterized weights are directly used to linearly interpolate 3D models.

The goal in this paper is that the performer utters some words, and the sculpted 3D models should be interpolated in a way as if it speaks.

Given the key-shape lips images $\mathbf{S} = \{S_1, ..., S_K\}$ and its corresponding 3D models $\mathbf{P} = \{P_1, ..., P_K\}$, a sequence of 3D facial animation can be built based on the original video sequence $\mathbf{F} = \{f_1, ..., f_n\}$. Key ideas are to parameterize video sequence using $\mathbf{S}$ into $\mathbf{W} = \{w_1, ...w_{18000}\} \in \Re^K$, transfer the coefficients to 3D model space, and use the coefficients and $\mathbf{P}$ to construct the corresponding sequence.

The first is to parameterize video sequence from image space into weight space $\mathbf{W}$ using $\mathbf{S}$, and is formulated as:

$$\min \| f_j - \sum_i^K S_i w_{ji} \|^2, \quad \forall f_j \in \mathbf{F} \tag{4}$$

The solution of this minimization is straight forward and a standard linear least square solver can apply. The obtained weight, though best fit for least error reconstruction, is not suitable for further editing and transfer. One important reason is over-fitting that when minimizing error, some weights will be much larger and others have to be negative to counter balance. If weight transfer is desirable then large or negative weight should be avoided. A non-negative least square (NNLS)[Lawson and Hanson 1974] solver is efficient and available that solve the following constrained minimization:

$$\min \| f_j - \sum_i^K S_i w_{ji} \|^2, \quad \forall w_{ji} \geq 0, \quad \forall f_j \in \mathbf{F} \tag{5}$$

The weight $\mathbf{W}$ can be directly used as α in Eq 2. Such direct transfer will, however, introduce an un-pleasing result because the non-linearity nature of projection as we introduce in Fig. 12. The observed feature points undergoes a projection that maps a *constant* angular movement to a *cosine* function, and while uniformly sampled on the angular space, the projection will result in a seemingly clustered sampled at degree 0. This non-linearity phenomenon is typically modeled using RBF, yet another alternative is proposed to model this projection.

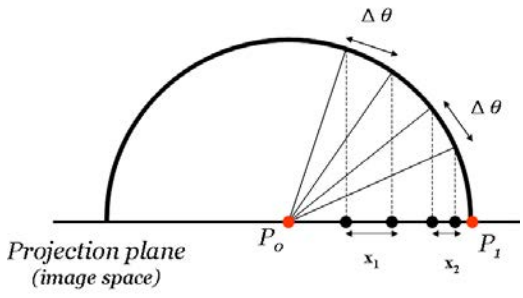

Figure 12: The projected point position x is the function of $cos(\theta)$, so the constant angular speed ω will result in a *sine* function, rather a constant speed for x.

One feasible solution is to select the corresponding features on the 3D model, blend them using $\mathbf{W}$ in the projection plane, i.e., x and y component only, and use these transferred projected points as constraints to plug in Eq.1 solving for α. Thus the result of 3D animation is $\alpha = \{\alpha_1, ...\alpha_{18000}\}$ defining the cross-mapped reconstruction parameters. Two primary reasons using the technique from Sumner et al.[Sumner et al. 2005] are the following:

1. Since the observation is two dimensional images, and by assuming fake orthogonal projection, the prior from examples and exponential-map provides fairly acceptable result when only two, rather than three, dimensional constraints are specified.

2. The use of exponential-map provides linearity that is useful for density estimation and multi-linear analysis. Take Fig. 12 for example: When the angular speed is constant, samples on angular space is uniform from $\{0...\frac{\pi}{2}\}$, yet when projected, over $\frac{2}{3}$, i.e. 60 degree, of samples are between $\{0, ..., \frac{1}{2}\}$ and leading a false density estimation.

The non-linear cross-mapping function using exponential-map gives an alternatives for mapping motion from one subject to another, while providing physical meaning of projection. Result of the 3D animation corresponds to the training video can subsequently used for estimating the probabilistic model phonemes.

5.4 Phoneme Space Construction and Trajectory Synthesis

After cross mapping a sequence of $\alpha = \{\alpha_1, ..., \alpha_{18000}\}$ associate with phoneme tag on each frame, the statistical analysis for phoneme set $\Phi = \{Ph_1, Ph_2, ..., Ph_{51}\}$ can be performed for density estimation:

$$\mu_{Ph_i} = \sum_j \frac{\alpha_j}{N_i}, \quad \Sigma_{Ph_i} = \sum_j \frac{(\alpha_j - \mu_{Ph_i})(\alpha_j - \mu_{Ph_i})^T}{N_i}, \quad \forall \alpha_j \rightarrow tag = Ph_i$$

where N_i is the number of training frame that correspond to the phoneme Ph_i, and the density is modeled with multi-dimensional Gaussian:

$$p_{\mu_{Ph_i}, \Sigma_{Ph_i}}(X) = \sqrt{\frac{1}{(2\pi)^K \|\Sigma\|}} \exp\{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)\}$$

Meanwhile, any speech $\{ph_1, ..., ph_T\}, \forall ph_t \in \Phi$ of length T tags can be synthesized with a sequence of $x = \{\alpha_1, ..., \alpha_T\}$ by the minimizing the following objective error function E consists of a data term and a smoothness term:

$$E = (x-\mu)^T D^T \Sigma^{-1} D (x-\mu) + \lambda x^T W^T W x \tag{6}$$

The data term is a measurement of normalized distance from each phoneme distribution center, and the smooth term prevent the path from being changing too excessively. In the objective function, x is a vertical concatenation of individual α_t at time stamp t, μ constructed from phoneme center that the speech is about to pass, and Σ from diagonal phoneme covariance, assuming independent in each dimension without loss of generality.

$$x = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_T \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_T \end{bmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \ddots & \\ & & & \Sigma_T \end{bmatrix}$$

D is a normalization term de-emphasizing longer duration phoneme that might influence the error minimization. W is a smoothness term that also models the co-articulation effects.

$$D = \begin{bmatrix} \sqrt{I - \frac{D_{ph_1}}{T}} & & & \\ & \sqrt{I - \frac{D_{ph_1}}{T}} & & \\ & & \ddots & \\ & & & \sqrt{I - \frac{D_{ph_T}}{T}} \end{bmatrix}$$

$$W = \begin{bmatrix} -I & I & & & \\ & -I & I & & \\ & & & \ddots & \\ & & & -I & I \end{bmatrix}$$

Taking the derivative of of Eq.6 yields the following equation that can be solved using standard linear system packages:

$$(D^T \Sigma^{-1} D + \lambda W^T W)x = D^T \Sigma^{-1} D\mu \tag{7}$$

The result from Eq. 7 synthesizes a path $x = \{\alpha_1, ..., \alpha_T\}$ that goes through each phoneme region spoken, and each time stamp α in the path can be plugged into Eq. 2 to get the 3D model and finally an animation.

The value λ plays a crucial role in the minimization.The physical meaning of the term can be interpreted as the degree of freedom to reach the steady state of a phoneme when speaking a word. If the value increased, the path tends to be smooth and the 3D model speaks as if he can not change the mouth too much. On the other hand when the value decreased, the models speaks likes a robot that would not be recognized as real person speaking but just remain fixed at each phoneme and changing shape suddenly at transition. This value is fine-tuned until appealing result is found, and current value is set to 10.

6 Result

Our testing platform is a laptop computer with Pentium-M 2.13GHz and 1.5GB RAM, where the testing 3D face model contains 3,201 vertices and 5,825 triangles.

The proposed framework is fast and storage efficient. To synthesize a path of a novel speech of about 10 seconds requires less than a few milliseconds. Given the non-linear blending parameters $\{\alpha_{i1}, ..., \alpha_{i7}\}$ for each frame i, the computation time required to solve Eq. 2 is 0.3 to 0.4 second. The density estimation for parameters α and Σ are 51 by 7 table of floating points each, and the examples required to synthesize animation are merely seven plus one for reference.

Currently the performance bottleneck lies in the term $M(F, \alpha)$ to blend feature vector non-linearly and the huge linear system of Eq. 2. Sumner et al.[Sumner et al. 2005] the authors suggest an improved Cholesky factorization by inspecting the sparse structure. To accelerate the linear system solver the performance of the function M, platform dependent Basic Linear Algebra Subroutine (BLAS) and storage optimization might apply to increase matrix/vector operation.

Figure 13: Top: Tracking video; Middle: Cross-mapped animation using the 7 key-shapes found in Sec 5.2; Bottom: Synthesized animation using utterances of the tracking video.

Fig. 13 shows certain frames of the tracked video, its cross-mapped animation, and synthesized result produced from the utterance independent of the tracking results. Comparing the animation, one can notice that the synthesized shapes may not be identical to the cross-mapped, which is made in correspondent to the tracking video and shall be used as ground truth. Inspecting the speech animation in the accompanying video, some artifacts can be observed when compared with the directly cross-mapped 3D animation. While largely recognizable of what is being spoken, it is still quite vague at some fine details at transition of mouth shape between open and close shape. Another un-natural feel is that the shapes of certain viseme do not reach the exact shapes as people expect, such as *'/b'* or *'/p'*.

We currently impute the phenomena to the design of synthesizing speech trajectory without preserving the sharpness. It might be solvable with more delicate design on the energy function, though still not known how to do at now.

7 Conclusion and Future Work

In this work a complete framework to synthesize speech animation of a subject given utterance is presented. 3D facial animation, traditionally considered as difficult to rig for motion, is automatically generated without the user even to touch model editing tools.

In this paper, affinity propagation is exploited to identify representative key-shapes that are used for 3D model creation, weighting space parameterizations, and new sequence synthesis.

A new method that originally used in mesh pose-editing is applied to weighting space parameterizations. While previous work primarily working on fine tuning un-intuitive kernel function parameters, the proposed parameterizations do not require any user intervention.

Finally a method, simple but without lose of generality, for estimating phoneme density is presented. Solving animation for subsequent novel speech is reduced as a trajectory synthesis problem that is easy to solve with standard linear system library.

The most tentative future is adding expression onto the face. While traditionally expression and speech are processed in separate, the two should be modeled in correlation.

Speaking without emotion is robotic and achieving no persuasion without expression. Currently bilinear analysis, with speech the content and expression the style, on two dimension image is well studied Tenenbaum and Freeman[Tenenbaum and Freeman 2000] and Chuang et al.[Chuang et al. 2002], Wang et al.[Wang et al. 2004] and multi-linear model by Vlasic et al.[Vlasic et al. 2005]. Applying bilinear analysis on 3D facial models is very tentative.

Currently the density estimation could be further improved by inspecting the property of speech. Although gaussian distribution is general for most cases, the data term measuring distance to the gaussian distribution center in trajectory synthesis might not truly reflect the way people speak. Because the density estimates samples from binary segmented speech, intermediate sample between two phonemes can be modeled with better techniques capturing such effects.

References

BARR, A. H. 1984. Global and local deformations of solid primitives. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 21–30.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 187–194.

BRAND, M. 1999. Voice puppetry. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 21–28.

BREGLER, C., COVELL, M., AND SLANEY, M. 1997. Video rewrite: driving visual speech with audio. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 353–360.

BUCK, I., FINKELSTEIN, A., JACOBS, C., KLEIN, A., SALESIN, D. H., SEIMS, J., SZELISKI, R., AND TOYAMA, K. 2000. Performance-driven hand-drawn animation. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, ACM Press, New York, NY, USA, 101–108.

CAO, Y., FALOUTSOS, P., KOHLER, E., AND PIGHIN, F. 2004. Real-time speech motion synthesis from recorded motions. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 345–353.

CHANG, Y.-J., AND EZZAT, T. 2005. Transferable videorealistic speech animation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 143–151.

CHOE, B., LEE, H., AND KO, H.-S. 2001. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation*, 2, 67–79.

CHUANG, E., AND BREGLER, C. 2002. Performance driven facial animation using blendshape interpolation. Technical report cs-tr-2002-02, Stanford CS.

CHUANG, E. S., DESHPANDE, H., AND BREGLER, C. 2002. Facial expression space learning. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, 68.

CHUANG, E. S. 2004. *Analysis, synthesis, and retargeting of facial expressions*. PhD thesis. Adviser-Christoph Bregler.

DENG, Z., AND NEUMANN, U. 2006. efase: Expressive facial animation synthesis and editing with phoneme-isomap control. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, 251–259.

DENG, Z., CHIANG, P.-Y., FOX, P., AND NEUMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, ACM Press, New York, NY, USA, 43–48.

EFTYCHIOS SIFAKIS, ANDREW SELLE, A. R.-M., AND FEDKIW, R. 2006. Simulating speech with a physics-based facial muscle model. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, 261–270.

EKMAN, P., AND FRIESEN, W. V. 1977. Manual for the facial action coding system.

EZZAT, T., GEIGER, G., AND POGGIO, T. 2002. Trainable videorealistic speech animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 388–398.

FREY, B. J. J., AND DUECK, D. 2007. Clustering by passing messages between data points. *Science* (January).

GEORGESCU, B., SHIMSHONI, I., AND MEER, P. 2003. Mean shift based clustering in high dimensions: A texture classification example. In *International Conference on Computer Vision*, 456–463.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1998. Making faces. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 55–66.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, ACM Press, New York, NY, USA, 399–407.

JONES, M. J., AND POGGIO, T. 1998. Multidimensional morphable models. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, 683.

LAWSON, C. L., AND HANSON, R. J. 1974. *Solving Least Squares Problems*. Prentice-Hall.

LEWIS, J. P., MOOSER, J., DENG, Z., AND NEUMANN, U. 2005. Reducing blendshape interference by selected motion attenuation. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, ACM Press, New York, NY, USA, 25–29.

NA, K., AND JUNG, M. 2004. Hierarchical retargetting of fine facial motions. *Comput. Graph. Forum 23*, 3, 687–695.

NOH, J.-Y., AND NEUMANN, U. 2001. Expression cloning. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 277–288.

PANDZIC, I. S., AND FORCHHEIMER, R. 2002. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. Wiley.

PARK, S. I., AND HODGINS, J. K. 2006. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics (SIGGRAPH 2006) 25*, 3 (Aug.).

PIGHIN, F., AND LEWIS, J. P. 2006. Facial motion retargeting. Siggraph 2006 course notes performance-driven facial animation, ACM SIGGRAPH.

PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI, R., AND SALESIN, D. H. 1998. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 75–84.

PYUN, H., KIM, Y., CHAE, W., KANG, H. W., AND SHIN, S. Y. 2003. An example-based approach for facial expression cloning. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 167–176.

SHOEMAKE, K., AND DUFF, T. 1992. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface '92*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 258–264.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, ACM Press, New York, NY, USA, 417–425.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, ACM Press, New York, NY, USA, 399–405.

SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, ACM Press, New York, NY, USA, 488–495.

TENENBAUM, J. B., AND FREEMAN, W. T. 2000. Separating style and content with bilinear models. *Neural Comput. 12*, 6, 1247–1283.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3, 426–433.

WANG, Y., HUANG, X., LEE, C. S., ZHANG, S., LI, Z., SAMARAS, D., METAXAS, D., ELGAMMAL, A., AND HUANG, P. 2004. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Computer Graphics Forum*.

WILLIAMS, L. 1990. Performance-driven facial animation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 235–242.

XIANG CHAI, J., XIAO, J., AND HODGINS, J. 2003. Vision-based control of 3d facial animation. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 193–206.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, ACM Press, New York, NY, USA, 548–558.