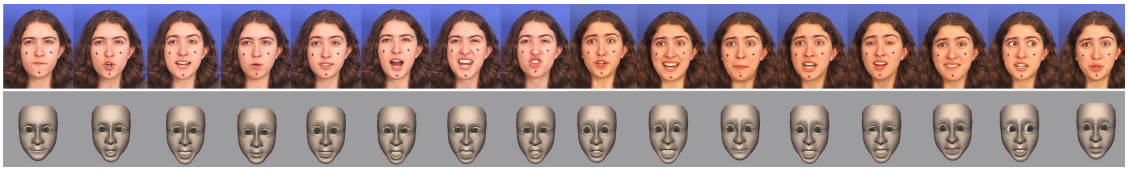
Performance Driven Facial Animation using Blendshape Interpolation

Erika Chuang *Chris Bregler*

Computer Science Department
Stanford University



Abstract

This paper describes a method of creating facial animation using a combination of motion capture data and blendshape interpolation. An animator can design a character as usual, but use motion capture data to drive facial animation, rather than animate by hand. The method is effective even when the motion capture actor and the target model have quite different shapes. The process consists of several stages. First, computer vision techniques are used to track the facial features of a talking actress in a video recording. Given the tracking data, our system automatically discovers a compact set of key-shapes that model the characteristic motion variations. Next, the facial tracking data is decomposed into a weighted combination of the key shape set. Finally, the user creates corresponding target key shapes for an animated face model. A new facial animation is produced by using the same weights as recovered during facial decomposition, and interpolated with the new key shapes created by the user. The resulting facial animation resembles the facial motion in the video recording, while the user has complete control over the appearance of the new face.

1. Introduction

Animated virtual actors are used in applications such as entertainment, computer mediated communication, and electronic commerce. These actors require complex facial expressions and motions in order to insure compelling interaction with humans. Traditionally, animators tediously hand crafted these animated characters. Recently, performance driven facial animation has been used to create these facial animations, greatly improving the efficiency of the animation process.

Advantages of performance driven systems include the speed up over manually crafted animations and the potential of producing more realistic facial motion. While many different techniques exist for performance driven facial animation, most methods track facial markers from an actor or actress, recover the 2D or 3D positions of these markers, and animate a 3D face mesh using the marker data. These methods typically require that the shape of the face in the source motion capture data closely resembles that of the target animated face. Otherwise, a method for retargeting the source motion data to the target face model is required.

There have been several different approaches to map motion data from the source to the target model. They generally fall into two categories: Parameterization, and Motion Retargeting. In parameterization, some universal facial parameters are derived from the motion data such that when they are applied to another face model, the parameters remain the same. However, simple parameters are often insufficient to describe more complicated facial expressions, while more complicated parameters are difficult to estimate from the motion capture data. In Motion Retargeting, the motion of one facial expression animation is mapped directly to the target mode. Since the target model may look different form the source, the source motion needs to be ‘adapted’ to the new shape.

There are many other popular methods for facial animation that are rarely used in conjunction with motion

capture system [Par96]. Blendshape interpolation is one example. Many commercial 3D animation software packages provide tools to make blendshape animations [Maya]. The principle of blendshape interpolation is similar to key shape interpolation. In this case, more than 2 key shapes can be used at a time, and the interpolation is for a single static expression, rather than across time. Each blendshape can be modeled using a variety of different methods. The amount of detail in each expression can also vary, as long as the resulting faces can be 'combined' in some manner. While this method is popular for specifying facial animation, it requires manual specification, and designing a complete animation can be quite time consuming.

We propose a method of making facial animation using a combination of motion capture data and blendshape interpolation. This method retains the flexibility of blendshape modeling, while gaining the efficient animation possible using motion capture. This technique consists of several stages: facial capture, facial decomposition and facial retargeting. An overview diagram of this method is shown in figure 1. In facial capture, we track facial features of an actress in a video recording using computer vision techniques. In facial decomposition, facial features are decomposed into a weighted combination of key shapes. These key shapes are automatically selected from the tracking data. Many existing tracking techniques use PCA to find a compact basis-shape set. Unfortunately, PCA based representations are not very well suited for our retargeting task. We present a new algorithm that discovers a better basis set. In facial retargeting, key shapes for the animated face model are created by the user. These key shapes for the new face model resemble the key shapes from the video sequence. For instance, a smiley face corresponds to a smiley face. Then, the decomposed weights for the key shapes in the video sequence are used to create interpolated facial animation for the new face model. We believe that even though the shapes of the face model have changed, the essence of the facial animation remain in the weights that transform the facial expression in between the different key shapes.

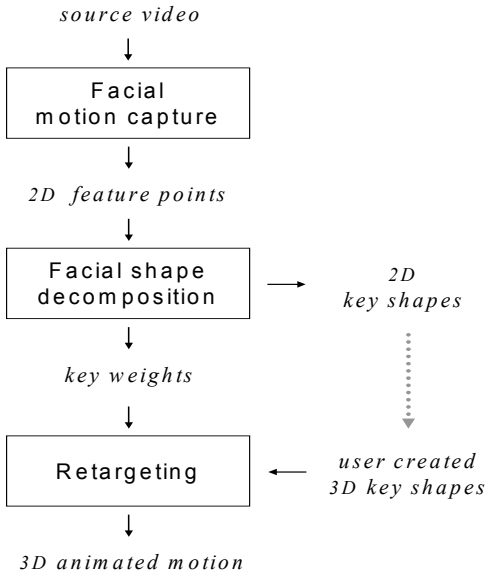


Figure 1: *Overview diagram of the Facial Animation System. A motion capture module extracts facial features from a source video sequence. The features are decomposed into a set of key shapes and key weights. The user creates an analogous set of key shapes in the target domain. These new key shapes, together with the extracted key weights are together used in a retargeting stage that creates the final animated motion.*

The advantage of this method is that the facial capture and retargeting are decoupled. The mapping from the source motion to the target is reduced such that only the weights for the blend shapes are transferred. Therefore, it is very easy to reuse a sequence on different models. The style of these models may vary greatly from the appearance of the original actress. Since many commercial tools already provide tools for blend shape interpolation, this method can be easily used in conjunction.

The outline of the paper is as follows: after a description of related work in section 2, we describe the facial motion capture method used in our system in section 3. Section 4 explains the process of decomposing facial expression into weighted combination of key shapes. In section 5, examples of 3D facial animation were created using the facial retargeting process. In section 6, we conclude our current work and discuss the future direction.

2. Related work

Most existing work on performance driven facial animation has focused on tracking and modeling of the human face [Gue99] [Wil90] [Mar00] [Ess96]. Through a 3D scanner [Cyb] or image measurement, a realistic face model is generated. The locations of the markers are used to drive the 3D model. Since the model usually has more details than the number of markers, a kernel function is typically used to deform the mesh so vertices that fall between the markers move properly. This work often assumes that the source and target face shape are identical.

When the target face model is different from the motion capture data, we must retarget the motion. There are two general categories as mentioned in the previous section. The first category involves some kind of facial animation parameters. The type of parameters can be simple distance

measurement, such as eye opening, mouth opening, and eyebrow raising etc [Buc00]. However, these parameters are often insufficient to describe more complex facial expressions. There is also work on estimating physical control parameters, such as muscle action, from image measurement [Ter93] [Ess96]. In general, muscle parameters are difficult to obtain from the motion capture because of skin movement, therefore these systems usually employ complex optimization process. There are also high-level qualitative expressions, such as happiness and sadness, or speech related parameters, such as visemes in various other systems [Har98] [Ost98]. In general, the higher the level of control parameters, the harder they are to estimate because of the weak link between the image measurement and the actual control parameters.

The other category of retargeting involves mapping motion vectors directly to another face model. Because the retargeted face model has different shape, the motion vectors are transformed to follow the curvature of the new shape [Lit94] [Bei92] [Noh01]. To retarget motion to a 3D model, this method requires dense mesh motion as input data, which may not be available from some motion capture systems. Furthermore it requires dense mesh correspondence, which may be difficult to specify when the source and target shapes are very different.

Our work is closer to the category of motion parameterization. Previous work has required an explicit parameterization and mapping that is often difficult to specify. This work uses an implicit mapping between models, embedded in the key shapes. We rely on the user's artistic skill to observe the source expressions and then model equivalent expressions in the target domain.

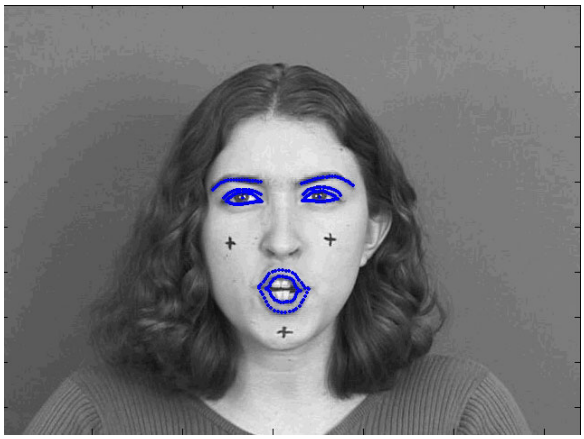


Figure 2: *Facial features are recovered using a tracking model trained on an annotated database of images.*

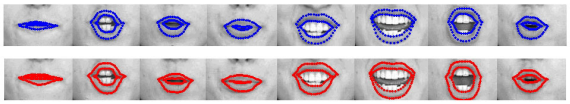


Figure 3: *Top – facial features for some example key frames obtained through tracking. Bottom – traced facial features for the same frames.*

3. Facial motion capture

The technique described in this paper requires a feature vector that describes an actor's motion. While a variety of motion capture techniques exist, the performance used here was recorded as a nearly marker-less face on a standard camcorder. Existing computer vision techniques were applied to track the facial features in this sequence. Features include the eyebrows, eyes, center of the pupil, and the inner and outer contour of the lips. The auxiliary marks drawn on the cheeks and chin where not used. We chose a model-based technique that can reliably track a marker-less face [Cov96]. We briefly describe the method here.

A tracking model is trained from an annotated face database. To obtain the database, a handful of images from the video sequence are first labeled with facial contours. Feature points are distributed evenly along each contour. The labeled images are then adjusted to remove the affine motion. Next, we perform image morphing between the pre-warped images to enlarge the image database. Principle Component Analysis is performed on a vector containing the pixel intensity values and the labeled feature locations of the face images in the enlarged database. The k largest eigenvectors are retained as the tracking model.

To track the facial sequence, the images are pre-warped to adjust for affine motion relative to the tracking model. Each frame is then projected onto the basis set of eigenvectors using only the pixel intensity values. Since the eigenvectors of the image database consists of both the pixel intensity values and the feature locations, the facial features of each frame can be obtained. Figure 2 shows a sample image of the tracked face.

4. Facial decomposition

The goal of facial decomposition is to take the facial features from the captured sequence and decompose them into a weighted combination of key shapes. The weights recovered during this process will be used to specify the animated motion of the target model. The key shapes are chosen via an automatic method that we will describe in section 4.2. These shapes should represent the variety of

facial expressions existent in the sequence. For example, figure 3 shows some example key shapes used for a video sequence in which only the lips were considered.

We define a key shape in the source sequence as the control points of the facial features in that frame, $S_i = [x_1,y_1,x_2,y_2,\ldots,x_n,y_n]^T$. The tracked facial features in each of the input frames can be described in terms of a weighted combination of a set of defined key shapes, that is,

$$S_t = F(\mathbf{w}, \mathbf{S}), \quad \mathbf{w} = [w_1,..,w_k], \mathbf{S} = [S_1,..,S_k] \qquad (1)$$

where F is the morphing function used to combine a set of key shapes, k is the number of key shapes, and the weight of key shape i is w_i. Facial decomposition is the inverse process of morphing. In image morphing, base images and weights are used to derive new warped images. In decomposition the morphing function and the resulting facial feature locations are known, but the weights must be recovered.

After determining key shapes and the associated key weights for a sequence, the motion can be retargeted. In the retargeting stage, the user creates a new set of face models that correspond to the key shapes from the input sequence. The new face models have facial features B_i, where the definition of B_i varies depending on the output model. Finally, the weights wi are applied to the output sequence

$$B_t = F(\mathbf{w}, \mathbf{B}), \quad \mathbf{w} = [w_1,..,w_k], \mathbf{B} = [B_1,..,B_k] \qquad (2)$$

The following sections describe two steps in the facial decomposition: determining the weights and choosing the key shapes. These two problems are not completely independent from each other. However, we will formulate them as separate processes in the paper for simplicity. Section 5 will describe facial retargeting and show some resulting output facial animations.

4.1 Determining weights

Given a set of input key shapes, $\mathbf{S} = [S_1, \ldots, S_k]$, finding the weights depends on the morphing function F. In this work we use linear morphing, that is,

$$S_t = \sum_{1}^{k} w_i S_i \qquad (3)$$

Weights that exactly satisfy this equation are unlikely to exist, but the best fit in a least square sense can easily be found. However, we are not interested only in a perfect reconstruction of the input facial features, we plan to use the recovered weights for retargeting. Minimizing the reconstruction error in the source sequence is not necessarily the most desirable thing. Our goal is to find the best weights to describe the process of creating each frame, as if it had been created using the morphing function and key shapes. Solutions that strictly minimize the least square error of equation (3) often introduce large positive and negative weights that counteract and compensate for one another. While these large weights minimize the error for the input shapes, they are not a good representation for describing the morphing process on a new target set of key facial features.

A simple test example illustrates the difficulty with merely minimizing error. Given a set of tracked lip contours as the key shapes in the source sequence, we traced these lip shapes by hand and used the control points along the traced shapes as the target key shapes, as shown in figure 3. Although the traced shapes closely match the tracked contours, they are not identical. Figure 4(a) shows one frame of the reconstruction of the lip contour in the source sequence using the weights from a least square solution. There are 21 key shapes used in this example. As expected, the reconstruction looks very similar to the tracked frame. Figure 4(b) shows the reconstruction when

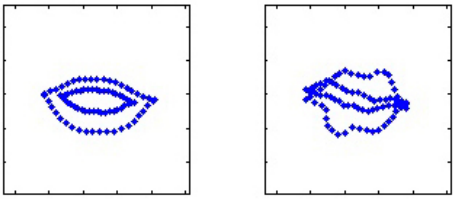


Figure 4: *Reconstruction of lip contours when the key weights were determined using a simple L2 norm. (a) Reconstruction in the original feature domain. (b) Reconstruction in the target domain. Even though the source and target key shapes were very similar, significant distortion is present.*

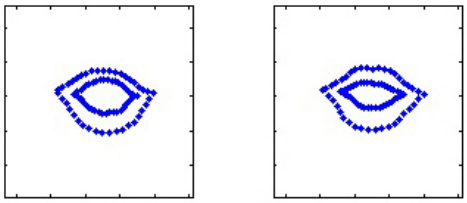


Figure 5*: Reconstruction of lip contours when the key weights were determined using a non-negative least squares fit. (a) Reconstruction in the original feature domain. (b) Reconstruction in the target domain. Note that the reconstructed shape in the target domain is faithfully reconstructed without distortion.*

the weights are used for retargeting onto the traced lip shapes. Even though the two sets of key shapes are very similar, the retargeted result has bad distortions.

Two simple methods can be used to minimize the distortion in the retargeted shape. First, the distortion reduces when the number key shapes are reduced, since this reduces the degrees of freedom. However, reducing the degrees of freedom also limits the amount of expressiveness in the animation and is not desirable. The second technique is to add constraints to the least square solution such that all the weights are positive. That is,

$$S_t = \sum_{1}^{k} w_i S_i, \quad w_i \geq 0 \tag{4}$$

This can be solved with standard numerical techniques [Law74]. Since negative weights are absent, large weights can not be counterbalanced. Thus, this constraint has the effect of reducing the number of large weights. However, it does not change the total number of key shapes, and thus preserves the expressive power of the representation. Figure 5 shows the same example of lip contour reconstruction and retargeting, where the weights were found using a non-negative constraint. The result has improved much throughout the entire sequence.

4.2 Choosing key shapes

Choosing appropriate key shapes is an important part of shape decomposition. Each key shape adds flexibility and expressiveness to the model, suggesting that many key shapes should be used. However, the user must create a target model for each key shape. In order to reduce user burden the number of key shapes should be kept small. An ideal method would balance these requirements to find the minimal set of key shapes that maintains the desired animation expressiveness. In this section we propose and evaluate several methods for choosing key shapes.

The problem stated above can be phrased in the following manner: Given a sequence of frames, each with its own facial shape, $\mathbf{Q} = [Q_1, Q_2, \ldots Q_t]$, we would like to pick k shapes from $\mathbf{Q}$ such that the rest of the sequence can be expressed as a combination of these key shapes with minimal error. Ideally this error would be measured in the target domain, since as discussed previously, even with low error in the source domain, retargeted shapes can become distorted. Unfortunately no ground truth data exists in the target domain, so we must resort to using reconstruction error in the source domain to evaluate the quality of sets of key shapes. Given a set of key shapes, $\mathbf{S}$, the best weight vector, $\mathbf{w}$, can be determined as above, and the error can be written as

$$E = \sum_{i} \| Q_i - F(\mathbf{w}, \mathbf{S}) \|^2 \tag{5}$$

Since finding the optimum set of key shapes is a large search problem that grows combinatorially with the number of key shapes, we instead propose three heuristic methods of choosing key shapes and evaluate them in terms of the metric given in equation (5). Each of these heuristics is designed to choose key shapes that cover the space of shape variation that exists in the sequence. Since the number of facial feature is relatively large, the space defined by the feature vector has a high dimension. Like any pattern classification problem, a smaller input dimension makes the problem more manageable. Therefore, all three methods make use of principle component analysis (PCA) to reduce the dimensionality of the input data.

Maximum spread along principle components. This method picks the data points that have the largest values when projected onto the principle axes. Starting from the first principle axis with the largest eigenvalue, the frames with the maximum and the minimum projection onto each axis are chosen as key shapes. Therefore, to pick k key shapes, we will use the first $k/2$ axes. The intuitive explanation of this heuristic is that for each direction of variation in the sequence, we want to choose the extreme maximum and minimum poses that encode this variation. All other shapes should be derivable as an interpolation between these two extremes.

Clustering. Each face shape is projected onto a small number of eigenvectors to produce a low dimensional vector. This low dimensional data is clustered into k groups using Euclidian distance. The center of mass of each cluster is chosen as a key shape. The intuition here is that by clustering the input data and choosing a key shape in each cluster, coverage of the entire range of motion is ensured. Clustering results are reported using both two and three dimensional feature vectors.

Convex hull. Each face shape is projected onto a small number of eigenvectors to produce a low dimensional vector. All shapes that lie on the convex hull of this space are chosen as key shapes. Since every other shape must lie within the convex hull, it is expected that all other shapes can be obtained by interpolating between the chosen key shapes. It turns out that for dimensions greater than two, nearly every frame of the example sequences lies on the convex hull. Since using every frame as a key shape defeats the purpose of this method, results are reported for the case of two dimensional feature vectors.

Figure 6 shows the reconstruction error of the control points around the lips as a function of the number of key

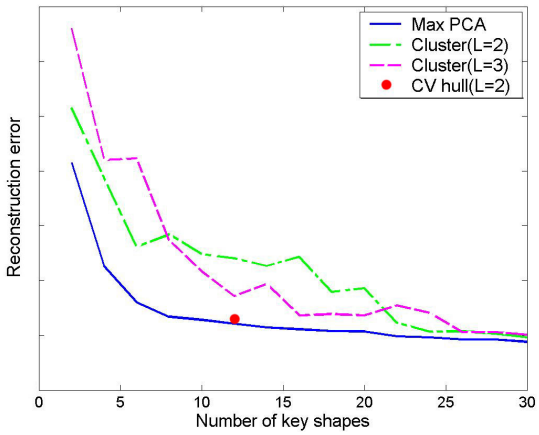


Figure 6: *Comparison of reconstruction error using key shapes chosen by different methods. The solid blue line is the error obtained when key shapes are chosen with the maximum spread in the direction of principle components. Note that this method consistently obtains the lowest error. The dashed green and magenta lines are the results of using clustering to choose key shapes. The red dot represents the error when key shapes are chosen on the convex hull of the feature space.*

shapes. The key shapes are chosen by the methods described above, while the weights are produced by least square with constraints as shown in equation (4). The solid blue line shows the error from using the maximum spread along the principle components. The dashed green and magenta lines show the errors from using the key shapes chosen by the clustering method. The green line is the result from using 2 principle components, and the magenta line is the result from using 3 principle components. Finally, the red dot shows the error from using the convex hull method. Since the number of data points on the convex hull is fixed, only one data point is present. Note that using the maximum spread along the principle components results in the smallest reconstruction error among the three methods. In addition, the error decreases monotonically as the number of key shapes increases, suggesting a graceful degradation when fewer key shapes are used. The errors produced by clustering also decrease with an increasing number of key shapes. However, the curves are somewhat noisy; and furthermore it is not clear whether using more dimensions is advantageous. The convex hull method seems to perform as well as choosing the maximum along principle components. However, since the number of chosen key shapes is fixed, this technique is less flexible.

From the above analysis, we concluded that the first proposed method, which picks the extreme data points along principle components axes, selects the key shapes that best represent the variety of shapes in the input sequence. This method also has the property that quality degrades gradually as the number of key shapes is reduced. Figure 7 shows some of the key shapes picked by this algorithm. Note that the set of key shapes consists of a variety of eye and mouth shapes, matching our intuitive notion of covering the space of possible poses.

5. Facial retargeting

To retarget the recovered facial motion to a new face model, the user creates new key shapes in the target domain. The weights that describe the combination of key shapes in each frame of the source sequence remain the same in the retargeted sequence. Only the key shapes themselves are replaced to form the new animation. Each frame in the new facial animation is a weighted combination of the new key shapes. It is important that the key shapes are consistent, for example if the mouth opens wider in key shape 3 than in key shape 5 and 6, the new key shapes need to have this property also. If the amount of 'extremeness' in the key shapes is contradictory, the output sequence will jitter in an uncontrolled fashion.

To demonstrate the facial retargeting, we use a 3D face model made with NURB surfaces. The output key shape vectors, $\mathbf{B} = [B_1,\ldots,B_k]$, described in section 4.2 now consists of the control vertices of the NURB surfaces. Other types of models may have different feature vectors. For example, for polygonal models, the vertices can be used as facial features. In order to minimize the number of

Figure 7: *Examples of key shapes picked by the proposed algorithm. Note that these shapes cover the space of variations existent in the recorded motion.*

key shapes, we divide the face into multiple regions. The eye region and the mouth region are modeled separately. The resulting faces are blended with linear weights in the intermediate region. The affine motion extracted during the facial capture stage is used to produce the global head motion in the retargeted sequence. We did not estimate 3D head motion, only scaled 2D translation and rotation are applied to the output motion.

The first video (included with submission) consists of a total of 386 frames, recorded at 30 frames per second. We used 10 key shapes for the eye region, and 12 key shapes for the mouth region. Figure 8(a) shows several sample frames from the input video sequence and the corresponding retargeting facial animation. Notice that the facial expression of the 3D model mirrors that of the input video sequence as desired. The second sequence shown in figure 8(b) consists of 396 frames. We used 8 key shapes for the eye region, and 14 key shapes for the mouth region. In this sequence, the motion of the eyes was also animated by treating the motion of the pupil as a separate region and modeled the eyes separately.

6. Discussion and future work

We have described a method of creating facial animation using a combination of motion capture and blendshape interpolation. A method for extracting a compact key-shape set that models motion variation was presented. Facial tracking data is then decomposed into a weighted combination of key shapes. These key weights encode the mouth shape, eye movements, motion, and style present in the original sequence. By transferring these weights to a new set of key shapes modeled by the artist, this characteristic motion is preserved. We show an example of this method in which the facial expressions of a talking actress in a video recording are retargeted to a different 3D face model.

The method described here is very intuitive and straightforward, and the resulting facial animation is expressive. Although we chose a particular method to capture the facial motion, and a particular 3D face model

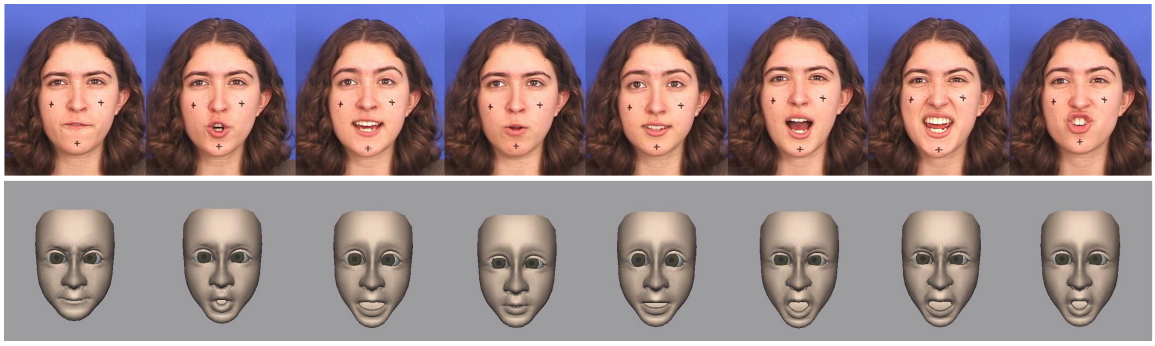


Figure 8(a): *Example frames from a video sequence that was retargeted onto a 3D face model.*

Figure 8(b): *Example frames from another sequence retargeted onto the same 3D face model. This sequence uses a new set of key shapes that better match the range of expression that exists in this sequence.*

for retargeting, this process is flexible and different methods and models can be applied. For example, the input face data could come from a commercial motion capture system. Similarly, the output face model could be a 2D drawing instead of a 3D model.

We are currently planning to investigate additional methods of face decomposition. The assumption of a linear interpolation function is limiting and could be generalized. In addition, better methods for extracting key shapes and determining key weights may exist.

Learning the dynamics of facial expressions is also an interesting future direction. Most existing techniques use raw marker positions as input training data. The learned dynamics can not be applied directly to a new face model without some method of re-mapping. It would be interesting to use key weights as input to a training algorithm. This may provide a more flexible model of motion dynamics.

References

[Bei92] Thaddeus Beier and Shawn Neely. "Feature-based image metamorphosis," Computer Graphics Proceedings of SIGGRAPH 92, 26 (2), pp. 35-42.

[Buc00] I. Buck, A Finkelstein, C. Jacob, A. Klein, D. H. Salesin, J. Seim, R. Szeliski, K. Toyama, *The First International Symposium on Non Photorealistic Animation and Rendering.* 2000.

[Cov96] M. Covel, C. Bregler, "Eigen-Points", Proc. *IEEE Int. Conf. On Image Processing, 1996.*

[Cyb] Cyberware, Montery, CA. *Head and Face Color 3D Scanner, Model 3030RGB/PS.*.

[Ess96] I. Essa, S. Basu, T. Darrell, A. Pentland, "Modeling, Tracking and Interactiv Animation of Faces and Heads using Input from Video", *Computer Animation Conference,* June, 1996.

[Gue99] B. Guenter, C. Grimm and D. Wood, "Making Faces", SIGGRAPH 1999 Proceedings.

[Har98] F. Hartung, P. Eisert, and B. Girod, "Digital Watermarking of MPEG-4 Facial Animation Parameters", *Computer & Graphics,* Vol. 22, No. 3.

[Law74] Lawson, C.L. and R.J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974, Chapter 23, p. 161.

[Lit94] P. Litwinowicz, L. Williams, "Animating Images with Drawings", SIGGRAPH 1994 Proceedings.

[Maya] Maya|AliasWavefront, Toronto, Ontario. *Maya Unlimited, V4.0,* 2001.

[Noh01] Noh, U. Neumann, "Expression Cloning", SIGGRAPH 2001 Proceedings.

[Mar00] S. R. Marschner, B. Guenter, S. Raghupathy, "Modeling and Rendering for Realistic Facial Animation", *The Eleventh Eurographics Rendering Workshop,* June, 2000.

[Ost98] J Ostermann, "Animation of Synthetic Faces in MPEG-4", Computer Animation, pp.49-51, Philadelphia, PA, June8-10, 1998.

[Par96] Parke and Waters, Computer Facial Animation, A.K. Peters, 1996.

[Ter93] D. Terzopoulus and K. Waters. "Analysis and synthesis of facial image sequences using physical and anatomical models." *IEEE Trans. Pattern Analysis and Machine Intelligence,* June, 1993.

[Wil90] L. Williams, "Performance-Driven Facial Animation", SIGGRAPH 90 Proceedings, 1990, 235-242.