Principal Component Analysis for Facial Animation

Khalid Goudeaux, Tsuhan Chen, Shyue-Wu Wang[+] and Jen-Duo Liu[+]

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213, USA
ksg@ece.cmu.edu, tsuhan@cmu.edu
http://amp.ece.cmu.edu
[+]Multimedia Lab. Institute for Information Industry, Taiwan

Abstract

This paper presents a technique for animating a three-dimensional face model through the application of Principal Component Analysis (PCA). Using PCA has several advantages over traditional approaches to facial animation because it reduces the number of parameters needed to describe a face and confines the facial motion to a valid space to prevent unnatural contortions. First real data is optically captured in real time from a human subject using infrared cameras and reflective trackers. This data is analyzed to find a mean face and a set of eigenvectors and eigenvalues that are used to perturb the mean face within the range described by the captured data. The result is a set of vectors that can be linearly combined and interpolated to represent different facial expressions and animations. We also show that it is possible to map the eigenvectors of one face onto another face or to change the eigenvectors to describe new motion.

1. Introduction

Animating a three-dimensional face is a difficult task due to the complexity of the human face. From a biological standpoint, there are over 20 superficial muscles that control facial expression plus the deep facial muscles that perform mastication and speech. Given such a real world representation, it has been demonstrated that a simplified muscle model can be applied to polygonal mesh models [1]. With this approach, several virtual muscles of the types linear, sphincter, and sheet are defined and assigned either a list of vertices to affect, a sphere of influence, or similar system of determining what vertices to influence. While this approach has the advantage of being intuitive, it is a tedious, manual process to create expressions and animate the model because the degree of contraction must be determined for each muscle by trial and error. In addition, the properties of each muscle are completely dependent on the face model with the effect that controlling a different face requires starting from scratch. Another disadvantage is that jaw movement does not easily integrate with the muscles. As the jaw rotates open in Figure 1, the muscles must also be translated, rotated and their contraction characteristics changed to keep the motion accurate. Otherwise, visual artifacts and strange facial contortions can occur. Lastly, the number of ways that the face can emote is limited to the number of expressions one is willing to implement. Working from the DEC-Face [1,2,3,4] application of the Cambridge Research Labs, pictured in Figure 1, we experimented with linear muscles with limited results. It was time consuming to place and calibrate the muscles and the facial motion frequently did not appear realistic. Our experience with the muscle-based approach led us to look for a more systematic approach to facial animation. Motivated by image-processing work done for motion segmentation and estimation [5] and by previous work in analyzing and synthesizing facial expressions [6], we moved to applying Principal Component Analysis [7,8] to animation parameters collected from a human subject and using the results for face animation. This paper specifies our method for describing facial motion more efficiently to allow complex facial animation and subtle facial expressions to be more easily and quickly obtained. In addition, we show that the collected animation parameters can be reused to animate new faces or altered to describe new motion.

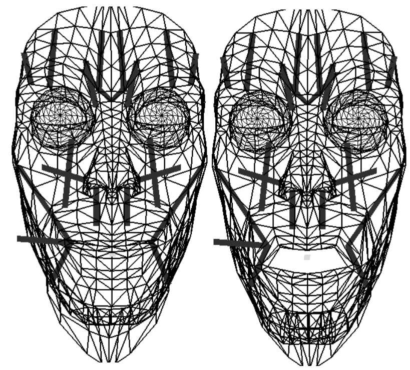

Figure 1 An Example of misaligned muscles

2. System Architecture

The system this paper presents relies on optical motion tracking to provide the training data for analysis. The principal components of data are found with PCA and then the face is interactively reconstructed based on the computed eigenvectors and eigenvalues. The three basic components of our system are Optical Tracking, Principal Component Analysis and the Graphical User Interface (GUI).

3. Optical Tracking

Facial feature tracking is done using the Real-Time HiRES 3D Motion Capture System by Motion Analysis [9]. Seven CCD high speed FALCON Video Cameras, pictured in Figure 2, capture the feature points at 240Hz. The cameras are connected to a 500 MHz Pentium II PC, with 250MB RAM, for collecting the data. Another PC installed with the MoCap Solver software to control the CCD cameras and analyzing the data.

A total of 39 facial features are tracked. A mapping has been created to show which points to link when visualizing the data so that the structure of the face is represented in a minimalist form as in Figure 3. The feature points were chosen to cluster more points around the mouth area where there are more complex facial muscles and motion. For example, around the mouth, there is the obicularis oris muscle that controls the lips and facilitates the complex lip motion for speech. The center of the nose, the eyes and the forehead are tracked using a minimal number of trackers. This number may be increased in the future to track facial expressions with greater precision.

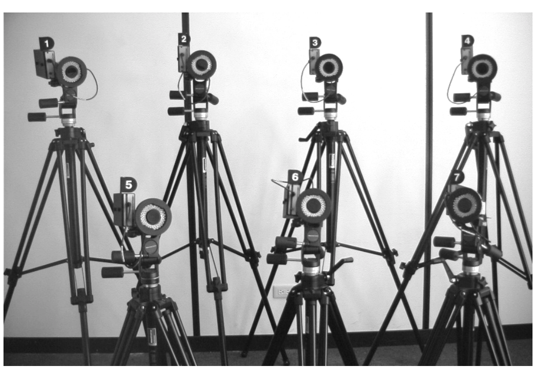

Figure 2 Infrared Motion Capture Cameras

4. Principle Component Analysis

PCA [7,8] is a statistical tool that decomposes data of high dimensionality to a set of orthogonal lower dimensionality vectors. The advantage of this technique is that if high dimensional data can be represented in a lower dimension then it can be more easily visualized, characterized or further analyzed. In the two-

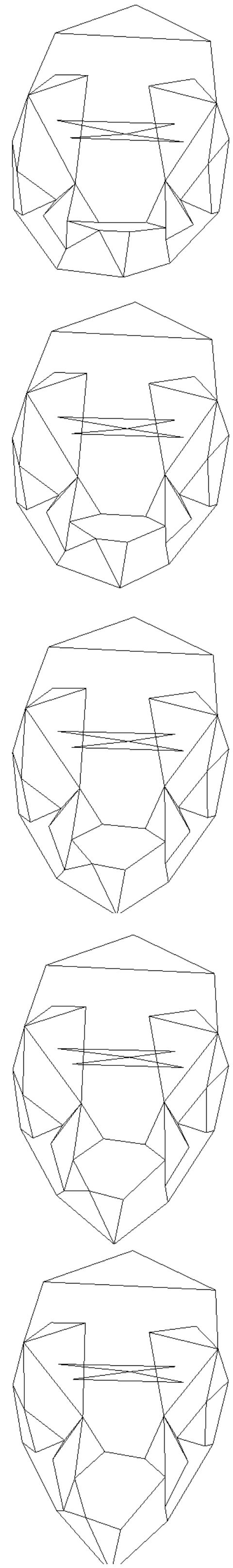

Figure 3 Opening the Jaw by combining principal components

dimensional case, a basic example of PCA would be a cloud of uniformly distributed points in the elongated shape of a football. The mean of the cloud would reside in the center of the football. The first principle component would be a vector originating at the mean and pointing along the long axis of the cloud; the axis with the greatest variance and the largest eigenvalue. The second principle component would also originate from the mean but would be orthogonal to the first component and have a lower eigenvalue. By linearly combining these two principle components and the mean, it is possible to represent any point within the cloud. For our captured data, this cloud has 39*3, or 117 dimensions.

For a given MxN matrix $\mathbf{A}$, PCA will decompose the matrix AA^T into a set of M eigenvectors of M dimensions and M eigenvalues as follows:

First, all of the frames in the sequence are averaged to compute the mean face vector $\mathbf{f}$. Next, for frame i of the length N sequence, the a 117x1 vector of the 39 feature points is constructed in the form:

$$\mathbf{v_i} = [x_1, y_1, z_1, \ldots , x_{39}, y_{39}, z_{39}]^T$$

This set of vectors is translated to the origin by subtracting $\mathbf{f}$ from each vector:

$$\mathbf{v_i' = v_i - f}$$

The set of all N vectors form the columns of the matrix $\mathbf{A}$:

$$\mathbf{A} = [\mathbf{v_1'}, \mathbf{v_2'}, \ldots , \mathbf{v_N'}]$$

PCA requires a square matrix to perform eigenanalysis:

$$\mathbf{B = AA^T}$$

The result of eigenanalysis is a set of 117 eigenvectors, $\mathbf{E}$, and eigenvalues $\mathbf{v}$:

$$\mathbf{E} = [\mathbf{e_1}, \mathbf{e_2}, \ldots , \mathbf{e_{117}}]$$

$$\mathbf{v} = [v_1, v_2, \ldots, v_{117}]$$

The eigenvectors in $\mathbf{E}$ can be linearly combined and weighted by a set of coefficients $\mathbf{c}$ to reproduce any of the frames of the original captured data and trivially create arbitrary new frames as follows:

$$\mathbf{c} = \{c_1, c_2, \ldots , c_{117}\}$$

$$V = \mathbf{f} + c_1 v_1 \mathbf{e_1} + \ldots\ c_k v_k \mathbf{e_k}$$

where k is an arbitrary number less than or equal to 117.

5. Mapping Eigenvectors

One of the goals of this research is to find a means for applying motion-captured data of one subject to another arbitrary subject. The two subjects need not appear similar geometrically. They would, however, need the number of control points and what they represent to correlate. A small step in this direction, displayed in Figure 6, is to allow a user to modify the mean face and the computed eigenvectors to create a new face that is controlled by an appropriately modified set of eigenvectors. More specifically:

Given $\mathbf{c}$, V, $\mathbf{v}$, $\mathbf{f}$, $\mathbf{E}$, $\mathbf{e}$ as above such that

$$V = \mathbf{f} + c_1 v_1 \mathbf{e_1} + \ldots\ c_k v_k \mathbf{e_k}$$

The user can interactively edit the mean face $\mathbf{f}$ and its set of eigenvectors $\mathbf{E}$ to produce $\mathbf{E}'$ and $\mathbf{f'}$ whose elements can be combined to produce V'

$$V' = \mathbf{f}' + c_1 v_1 \mathbf{e'_1} + \ldots c_k v_k \mathbf{e'_k}$$

6. PCA Face Application (GUI)

This GUI, shown in Figure 4, is implemented on the Windows platform as an MFC application rendering an OpenGL scene. A given input data set is read from a text file and analyzed. Its mean, eigenvectors and eigenvalues are stored and the mean face is displayed. The user can view an animation of the original captured data or vary the weighting of the principle components to create various facial expressions. Two versions of the face are displayed. The face on the left shows the mean face directly resulting from PCA analysis and is controlled by the corresponding set of principal components. The face on the right can be edited in two ways. The mean can be edited to create a new face and the principal components can be individually edited to provide more appealing control of the new face. The original PCA data provides a useful starting point for controlling new face models.

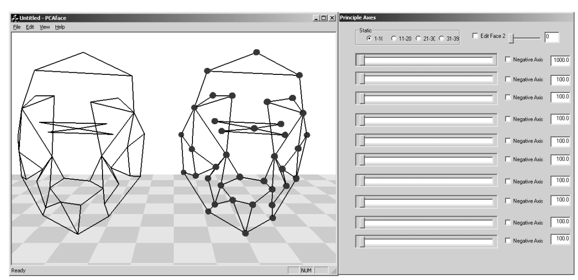

Figure 4 OpenGL Application

7. Results

The captured test data varies from 300 to 2000 frames at 15 frames per second of a man speaking normally. By visual inspection, the motion with the largest variance in the sequence is the opening and closing of the jaw. From this, it is expected that the first principle component would open and close the mouth and deform the cheeks accordingly.

As expected, the first few components affected the jaw as shown in Figure 5. The first component caused the lips to protrude and the mouth to open and also widen slightly. The second component moved the jaw in a lateral motion as the speaker frequently moved his jaw laterally when speaking. The third component also

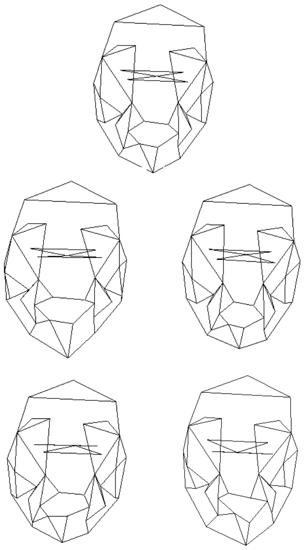

Figure 5 Top to Bottom: mean face, adjusting 1st principal component, 2nd principal component

opened and closed the mouth slightly while also exhibiting the lateral jaw movement. By moving these first three components it is possible to open and close the mouth, as in Figure 3.

8. Conclusions and Future Work

This paper has presented an application of Principal Component Analysis to Facial Animation. Using real data captured from a live actor, the system can allow a user to manipulate a 3-dimensional face in real time and manipulate the set of control vectors for that face. We feel that PCA is a promising means to taming complex facial animation as demonstrated by our ability to open and close the mouth by varying one parameter. Currently we are planning to capture more test data, a process that is prolonged by the expense of optical motion capture hardware. Our current data sets are of a person speaking and so we plan to acquire new datasets of a person changing facial expressions.

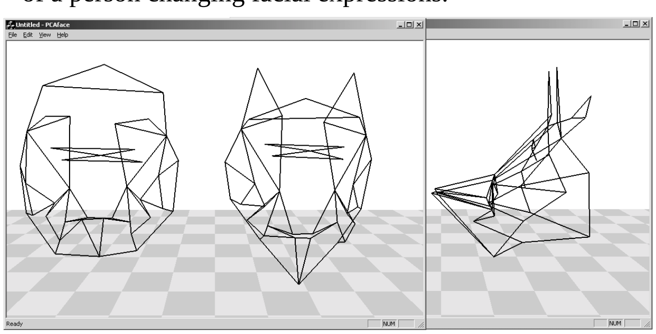

Figure 6 Modified eigenvectors closing the mouth on a new face (an ugly fox)

We would like to develop a mapping routine to use the captured data to manipulate an arbitrary face. Currently, the user can manually describe a mapping by altering the mean face and its eigenvectors. In light of recent research on the subject of retargeting motion captured data[10,11], we feel that with PCA, much of the process can be automated.

We have been exploring more intuitive ways for a user to interact with the face model. One interesting avenue of control is puppetry[12]. Providing the user with some physical input device, the user could interactively animate the face and ideally, effectively emote through the face. One interesting application is adding new frames to existing animations via interpolation or creating entirely new frames. As Figure 6 demonstrates, modifying the data set in the eigenvalue space to create smooth in-between or arbitrary new frames is simple.

9. References

[1] Parke, F.I., Waters K. ***Computer Facial Animation***. A K Peters, Wellesley, MA 1996.

[2]K. Waters. ***A muscle model for animating three-dimensional facial expressions***. Computer Graphics (SIGGR.4PH’87), 21(4):17-24, July 1987.

[3] Keith Waters and Thomas M. Levergood. ***DECFace: An automatic lip-synchronization algorithm for sythetic faces***. Technical Report CRL 93/4, Digital Cambridge Research Laboratory, September 1993.

[4] Y. Lee, D. Terzopuolos, K. Waters, ***Realistic Modeling for Facial Animation***, Proc. SIGGRAPH 1995, pp. 55--61, Aug. 6--11, Los Angeles.

[5] Y.T. Lin, Y.K. Chen and S.Y.Kung, ***A Principal Component Clustering Approach to Object-Oriented Motion Segmentation and Estimation***. Journal of VLSI Signal Processing 17, 1-25, 1997.

[6] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin, ***Synthesizing Realistic Facial Expressions from Photographs***, Siggraph proceedings, 1998, pp. 75-84

[7] I.T. Jolliffe, ***Principal Component Analysis*** Springer Verlag, 1986

[8] M. Kendall. ***Multivariate Analysis***. Charles Griffin&Co., 1975

[9] Motion Analysis - www.motionanalysis.com

[10]M. Gleicher, ***Retargeting Motion to New Characters***. Proceedings of the 25th annual conference on Computer Graphics, 33-42, 1998.

[11] Andrew Witkin and Zoran Popovi'c. ***Motion warping***. *Computer Graphics Proceedings*, pages 105--108, August 1995

[12] S. Sidney Fels and Geoffrey E. Hinton. ***Glove-TalkII: A neural network interface which maps gestures to parallel formant speech synthesizer controls***. IEEE Transactions on Neural Networks, 9 (1998), No. 1, 205-212.