A Rapid Face Construction Lab

Akikazu Takeuchi and Steven Franks

SCSL-TR-92-010

May 7, 1992

Sony Computer Science Laboratory Inc.
3-14-13 Higashi-gotanda, Shinagawa-ku,
Tokyo, 141 JAPAN

Copyright © 1992 Sony Computer Science Laboratory Inc.

A Rapid Face Construction Lab

Akikazu Takeuchi and Steven Franks
Sony Computer Science Laboratory Inc.
Takanawa MUSE BLDG.
3-14-13, Higashi-Gotanda,
Shinagawa-ku, Tokyo 141
Japan
{takeuchi,stevef}@csl.sony.co.jp

1 Introduction

Faces are the most often used, the most expressive and the most sophisticated interfaces of humans. A well synthesized face may induce the illusion that it is alive. This leads to the concept of *Virtual Humanity*. A virtual human is a computer generated human model which can be experienced by everyone and give the feeling of contacting a real human. The concept of virtual humanity is different from Artificial Intelligence in the respect that it focuses on emotional behavior of humans. As we are moved by actors and actresses of good movies, these virtual humans will move us, hence they could open a new communication channel between a computer and a human. From the viewpoint of long-range research, this new research will be a basis for the development of software which moves people impressively and profoundly.

Currently we are intensively investigating facial expression. This is because a face is the most expressive and hence cognitively important part in a human body. It often is regarded as symbolizing an owner's personality. Namely *face* $\approx$ *personality*. Hence to synthesize a face and its expressions is directly connected to synthesis of humanity.

Currently we are concentrating on modeling and facial expression, and are developing an environment for rapid face construction. Its purpose is to support various experiments with synthesized faces. The environment is composed of three systems:

1. 3D Modeling System
2. Animation/Rendering System
3. Script Control System

The systems are modular and can be easily interchanged for experimentation purposes. The overall system architecture is described in Figure 1. In the following, Section 2 describes 3D modeling system and Section 3 describes Animation/Rendering System and Script Control System.

2 Face from Video: 3D Modeling of A Human Face

A popular method of 3D face modeling is based on photographs of a real face with a mesh on its surface. These photographs are processed to extract 3D coordinates of the face surface. This method is not applicable when a person to be modeled is not available because of, for example, his or her death. In such case, the plaster model of the person's face is used. However, construction of plaster models is expensive and also time consuming.

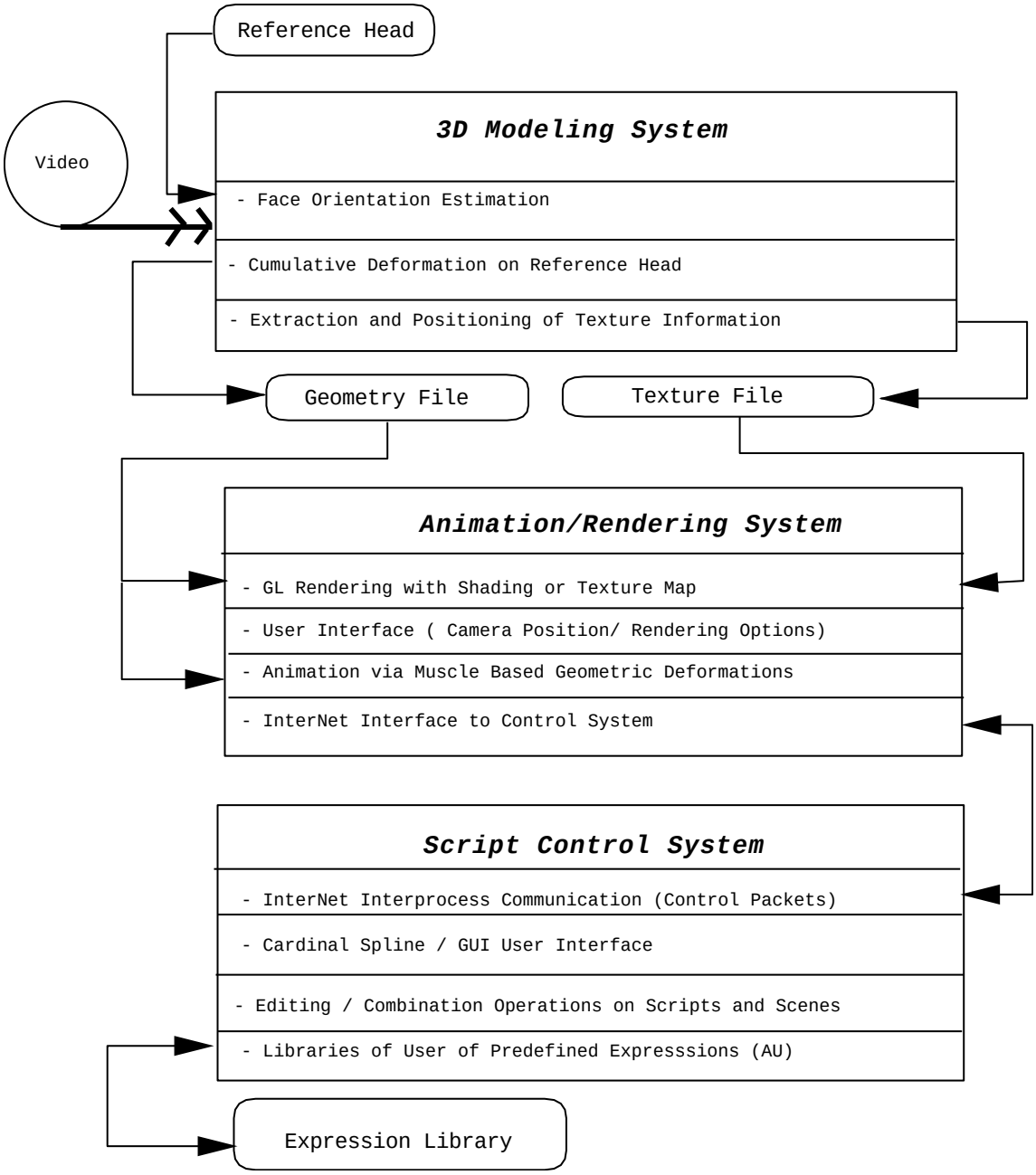

Figure 1: Configuration of A Rapid Face Construction Lab

Methods using advanced measuring technologies such as laser range scanner of Cyberware, Inc., and a 3D digitizer of Prohimus, Inc., are now becoming popular. However, these have the same limitation as the photo-based method, that is, a 3D model such as a plaster model is required when modeling a dead person.

We are developing a new method overcoming this limitation and enabling 3D modeling of any person we know. Our method is **3D modeling from unrestricted pictures**. Unrestricted pictures include video clips, a sequence of photographs and films. Our method is supported by technologies that have been developed in an active field in computer vision research, "3D recovery from 2D image." "Face from video" is a special case of "3D from 2D". This restriction can also be applied to make our method a more expectation-based system, which means that in many situations the knowledge that a target is a human face is utilized to reduce ambiguity of recognition. This approach is generally regarded as *model based* (model expecting) pattern recognition.

"Face from Video" also expresses our claim that insists on shallow inferences from various aspects on many pictures rather than deep inference on a few pictures. The basic principle of our modeling system is that a model can not be completely constructed from one picture. We achieve a better model by refining the model iteratively, with a sequence of pictures. We do not spend much time for one picture. Rather we want to examine more pictures to refine the model. Even if information extracted from each picture is limited, the modeling process as a whole converges. This is because a sequence of pictures provides a new picture for every frame and provides new information to the modeler.

More concretely speaking, our approach is similar to step by step refinement starting from some initial model. In each step, the resultant model of the previous step is used as an approximated model of the true shape, and various geometrical parameters are computed based on that, hence the model is refined.

Modeling starts with an initial model, which is an arbitrary 3D face model and is called *a reference model*. For the sake of simplicity, currently it is assumed that a face is symmetric. For each picture, the reference model is used when interpreting a 2D image of a face. The interpretation is based on making the best overlapping between the 2D image and the 2D image of the reference model projected after appropriate transformations (translation and rotation). When comparing overlapping two images, differences from various viewpoints may be discovered. Then, the reference model is modified to eliminate these differences, and the modified reference model is used in interpretation of the next picture.

In computer vision research, various information is considered to help 3D reconstruction. They are, contour, shading, texture, motion and so on. We are currently using contour, and considering the use of shading information.

Modeling Process

For each member (not necessarily for each frame) of a picture sequence, the modeler performs the following semi-automatically and deforms the reference model to be closer to the face in the picture. A face looking straight is called a face in *the normal position*. Figure 2 shows a 3D-coordinate system for the reference model in the normal position. Orientation of a face is defined to be amount of rotation applied to the face in the normal position. Here we describe the outline of processing.

1. Extraction of a face
   A region which corresponds to a face is extracted. The only operation required is posi-

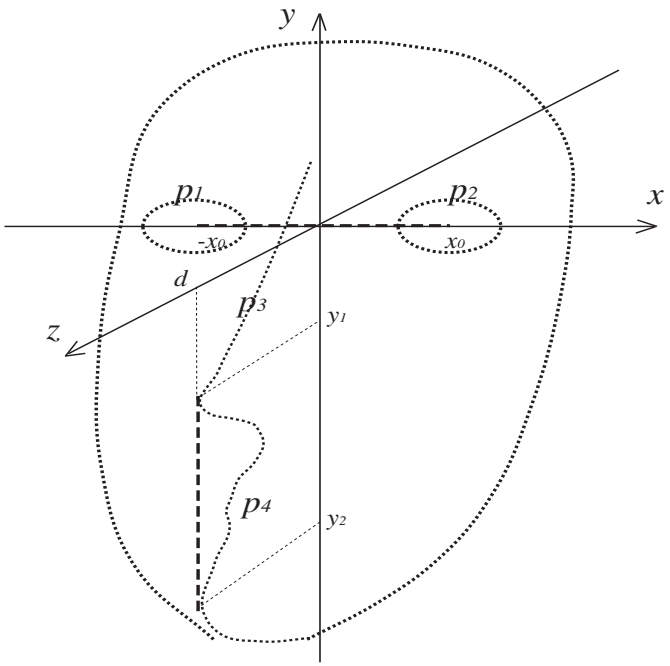

Figure 2: A Reference Model and Its Associated 3D Coordinate System

tioning of four points in a face. They are centers of both eyes, the top of a nose, the top of a chin. Currently this is done with user assistance. These four points in a face are selected for the estimation of orientation. These points are advantageous because they correspond to features in a face which are easy to find and their geometric relation can be represented by only four shape parameters, x_0, y_1, y_2, d (See Figure 2). The more points we mark, the more information we have, but at the same time the more noise we have so that the estimation error of the orientation increases.

2. Determination of orientation of the face & Macroscopic deformation
An orientation to which the face directs is estimated by analyzing 2D geometrical relations among the four feature points above.

   Let p and p' be points in the reference model in the normal position and points in its rotated model, respectively. Given a 2D image of the face, the problem of determination of orientation of the face is to determine the rotation matrix R such that

$$p' = Rp \tag{1}$$

   where the 2D projection of p' best fits to that in the 2D picture. Matrix R has nine elements, but since it is an orthogonal matrix ($R^tR = I$) the degree of freedom is three. Four feature points p_i are adopted for solving Equation 1. Note that 3D coordinates of feature points in the reference model in the normal position are:

$$\begin{aligned} p_1 &= (-x_0\ 0\ 0)^t \\ p_2 &= (x_0\ 0\ 0)^t \\ p_3 &= (0\ y_1\ d)^t \\ p_4 &= (0\ y_2\ d)^t \end{aligned}$$

where $x_0, d > 0, y_1, y_2 \leq 0$ (See Figure 2).

Let $1/s$ be a scaling factor such that $p_i' = sp_i''$, and p_i'' be $(X_i\ Y_i\ Z_i)^t$. By solving the above equation,

$$R = \begin{pmatrix} \frac{s}{x_0}X_2 & \frac{s}{y_1-y_2}(X_3-X_4) & \frac{s}{d(y_1-y_2)}(y_1X_4-y_2X_3) \\ \frac{s}{x_0}Y_2 & \frac{s}{y_1-y_2}(Y_3-Y_4) & \frac{s}{d(y_1-y_2)}(y_1Y_4-y_2Y_3) \\ \begin{vmatrix} \frac{s(X_3-X_4)}{y_1-y_2} & \frac{s(y_1X_4-y_2X_3)}{d(y_1-y_2)} \\ \frac{s(Y_3-Y_4)}{y_1-y_2} & \frac{s(y_1Y_4-y_2Y_3)}{d(y_1-y_2)} \end{vmatrix} & \begin{vmatrix} \frac{s(y_1X_4-y_2X_3)}{d(y_1-y_2)} & \frac{sX_2}{x_0} \\ \frac{s(y_1Y_4-y_2Y_3)}{d(y_1-y_2)} & \frac{sY_2}{x_0} \end{vmatrix} & \begin{vmatrix} \frac{sX_2}{x_0} & \frac{s(X_3-X_4)}{y_1-y_2} \\ \frac{sY_2}{x_0} & \frac{s(Y_3-Y_4)}{y_1-y_2} \end{vmatrix} \end{pmatrix} \quad (2)$$

Assuming orthogonal projection, X_i, Y_j $(1 \leq i, j \leq 3)$ are obtained as the 2D coordinates of the corresponding points in the 2D image. Hence, using Equation (2), R can be calculated directly from the 2D image.

When a face model is being constructed, final values of its shape parameters are not known. In our scenario "Face from Video", orientation determination is done to compare the current reference model with the target face in the picture from the same view angle. That means orientation estimation intrinsically has to work under inexact shape parameters and even with an inexact scaling factor.

Inexact shape parameters and an unknown scaling factor mathematically mean that a matrix given in Equation (2) is not necessarily an orthogonal matrix. Since values, X_i, Y_j $(1 \leq i, j \leq 3)$, are directly obtained from 2D images under orthogonal projection, there are five variables, s, x_0, y_1, y_2, d in Equation (2). Thinking conversely, we can determine those variables so that R satisfies the constraint $RR^t = I$. New shape parameters obtained by solving this equation can be applied to deform the reference model. The deformation is macroscopic since it only changes the ratios among four parameters.

Once an orientation is obtained with a scaling factor and updated values of shape parameters, the reference model is scaled, deformed and rotated in the same orientation so that its projection to xy-plane overlaps best with the face on the 2D picture (See Figure 3).

3. Contour fitting

Comparison between a contour of the reference model and that in the picture is performed. If there is a gap, the reference model is modified so that its contour fits that in the picture.

The deformation is performed in the same way as deformable models by Terzopoulos *et al.* [Terzopoulos *et al.* 88]. In this model, polygons constituting a face is considered as a node/spring system. Motion of each node is governed by the following Laglangean equation:

$$m_i \frac{d^2\mathbf{x}_i}{dt^2} + \gamma \frac{d\mathbf{x}_i}{dt} + \mathbf{p}_i + \mathbf{q}_i = \mathbf{f}_i \quad (3)$$

where $\mathbf{x}_i$ is a 3D coordinate of the i-th node, m_i a mass, γ a dumping factor, $\mathbf{p}_i$ net spring force, $\mathbf{q}_i$ smoothness constraint, $\mathbf{f}_i$ image force. Image force is turned on only on the nodes close to the occluding boundary. The equilibrium is achieved when image force (stronger in an edge) to a node near the boundary balances with spring force and the smoothness constraint. We are currently experimenting with various parameters and various smoothness constraint.

Note that, since a face is treated as a symmetric object, deformation applied to one side of the face influences the other side.

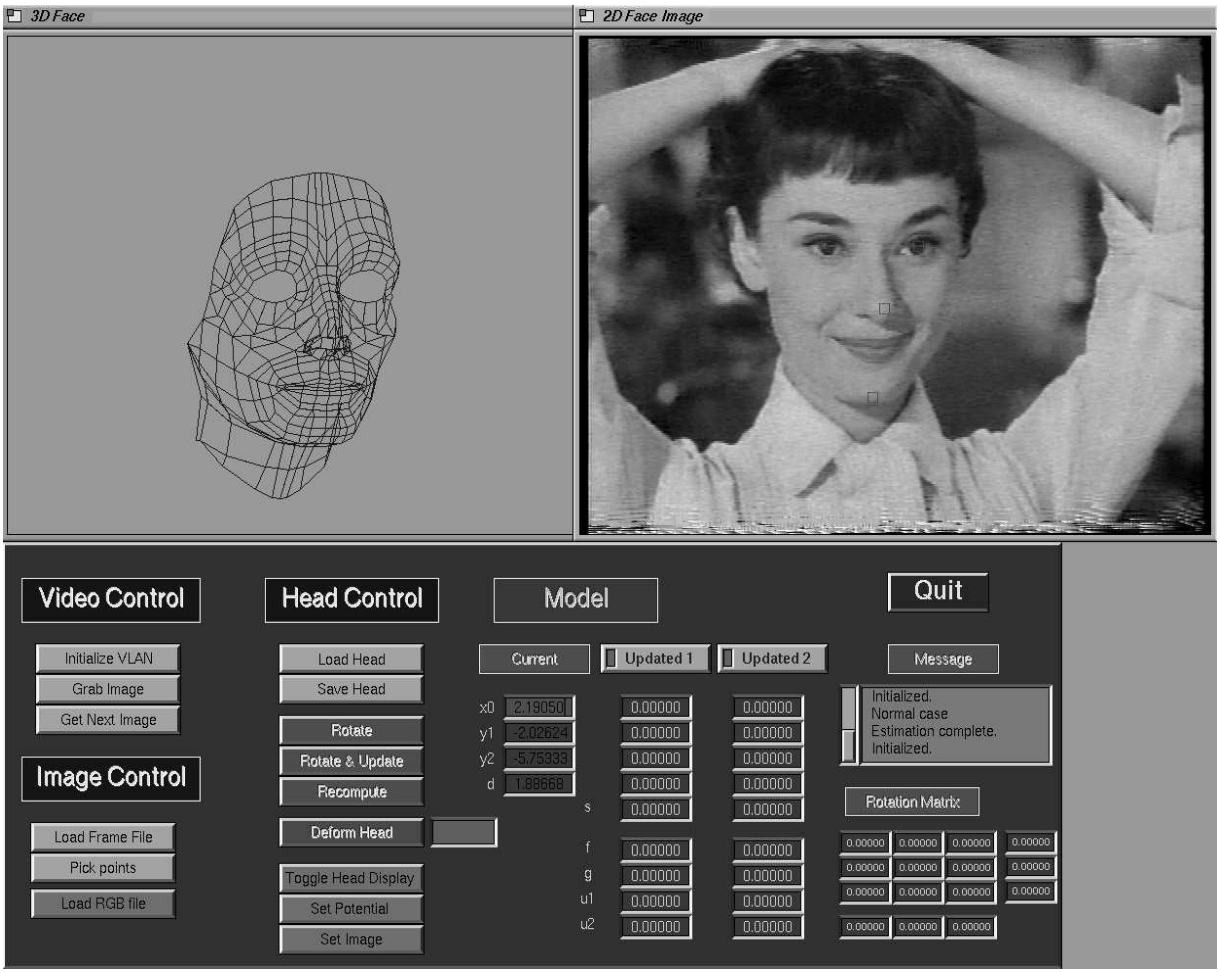

Figure 3: 3D Modeling System: A bottom window is a control panel of 3D modeling system. It control a 2D image window (upper right), a reference model window (upper left), and input resources such as rgb files, image grabber connected to a VCR. It monitors the orientation estimation and allows user intervention when necessary. In a 2D image window, a picture from a movie "Roman Holiday" is shown with four feature points marked. In a reference model window, the reference model after the application of the estimated rotation and deformation is shown.

4. Texture mapping
   One advantage of this process is that, even if the reference model does not perfectly fit to the target face in the picture, we can cut out the facial texture from the picture and map it to the reference model. Since the reference model and the picture fit exactly in the four points (both eyes, nose, chin), the mapped texture almost fits to the reference model. Figure 4 shows the reference model with a texture obtained in Figure 3. Although small displacements around eyes and a mouth are inevitable and sometimes they spoil the overall impression, we are considering to remedy those by applying the same deformation technique as the above. That is, in addition to near boundary node, we make nodes around the eye hole and nodes on a lip edge sensitive to image force so that they can automatically seek eyes and lips.

Resultant geometry data of the reference model and the texture data are then written to disk for use by the Animation and Rendering System.

3 An Interactive Facial Animation Testbed

3.1 A Modular Distributed Architecture

To facilitate experimentation with a variety of scripting and animation methods, the animation, rendering and control sub-systems were implemented in a highly modular fashion. Currently, the Animation and Rendering Systems (ARS) are one process, which communicates with the Control process via a Internet communications socket. The ARS can also run in stand-alone mode by reading parameters from a disk file. All facial and texture data are stored in disk files, allowing different heads and textures to be animated. Figure 1 illustrates the basic structure of the system. In addition, the animation and control systems were designed to exploit coarse grain parallelism. It is possible to run the ARS and Control modules on different machines, or at different times. This approach allows for easy interchange of a variety of control and animation systems. Currently, a muscle based deformation animator and spline-based control system are being used. The interactive spline control system provides a flexible and powerful script editing environment where facial motion can be described in terms of predefined action units which are converted to spline curves. Users can define additional action units and make detailed modification of all action units inserted into the animation script. A variety of editing functions are implemented on the spline based scenes.

3.2 Animator/Renderer System (ARS)

Our current facial animation system is based upon the deformation equations of Keith Waters [Waters 87] [Wang 92]. Deformations are applied on a set of points lying on the surface of the face. The deformations simulate the actions of various facial muscles. Jaw rotation and eye movement are also incorporated. The face is composed of approximately 500 points.

The program initially loads all of the point and polygon data from disk. This baseline state corresponds to a face in a "relaxed" position. For each frame, a series of muscle deformations are applied to the points, in the baseline position, after which the face is rendered. The degree to which each muscle deformation is applied is controled by muscle "activity" parameters. These can either be read directly from disk or obtained via an Interprocess Communication Socket from the Control Editor.

Figure 4: Interactive Animation Environment: On the right of the screen are splines which control the muscle activation levels of the face. The user manipulates the scripts by moving the spline control points or selecting operations from the Muscle Control Panel (Lower Left). The Upper Left portion of the screen contains the Animation and Rendering System which provides neasr real-time interactive feedback to the script designer. This example uses a texture map and facial geometry obtained from the 3D Modeler.

The face may be rendered using a skin-like surface material with Goaurard shading or by applying a texture map generated by the the technique decribed in Section 2. Head orientation is controlled by the animation sub-system parameters, and the user can interactively control the "camera" position from which the head is viewed.

Base performance ranges from 20-25 fps on an SGI Power Series. Optimizations have not yet been implemented, but will be discussed in the full paper.

Also under development is a physically based facial animation system controlled by muscles. This system is controlled by the same parameters as the deformation based animator, and can easily be interchanged. The main benefit of this method being the ability to generate wrinkles automatically, it is also more robust in the face of widely varying head geometries. However, our model currently appears too slow for interactive use.

3.3 Control System

A major portion of our research is devoted to exploring different control methods. We have chosen an animation system controlled by muscle value parameters because of their universality. This allows an "open" interace standard which enables different animation and control systems to be easily interchanged. It also allows us to draw upon an extensive body of existing psychological research. Higher level abstractions can easily be placed on top of the muscle level, such as Eckmann's Action Units [Ekman and Friesen 77].

Our control system is designed to be highly interactive. The user must be able to precisely define muscle parameter values, while being able to access higher level abstractions. We have developed a system based upon Cardinal Splines which fulfills both these requirements.

The system is implemented in an Object-Oriented fashion using C++. The core class being a *spline editor* which allows a user to place an arbitrary number of Cardinal spline control points on a two dimensional grid. The x-coordinate of the spline corresponds to time, while the y-coordinate corresponds to the level of muscle activity. A variety of functions are available to manipulate the spline editors; scaling in time, scaling in intensity, and an assortment of combination functions. A *script* is composed of spline editors, each of which are associated with one or more muscle parameters. The script defines the current length of the animation. At any time the user can preview the script frame-by-frame or by animating a specified duration. The script can be modified either by directly manipulating the control points in the spline editor or by prepending, appending, or combining additional scenes. *Scenes* are a collection of spline editors, similar to a script. The major differences are that a scene need not have spline editors for all the animation parameters, and is usually defined over a shorter period of time. Scenes can be saved/loaded from disk, and incorporated into the script in various ways. Scenes are used to define common action units (AU). We are investigating combining emotions in a way that the essence of each emotion is preserved as much as possible, rather than just interpolating between emotions which occur simultaneously. We feel spline are a good base to build a control system on because of the ease of combining spline segments and their inherent property of smoothness. They also provide a compact data representation for storing detailed information. The current system allows the user to load predefined Action Units (scenes) and subsequently "fine tune" the animation with real-time interactive feedback from the Animation/Rendering System.

Further research will focus on the psychological aspects of facial animation. In particular, what key features of an expression does the human mind use to classify emotional expressions. This is an active area of research in the areas of neuropsychology and ecological psychology.

Currently the FACS notation is widely used, and while very helpful, it is basically a static description of facial expressions. By drawing upon this work we hope to get a better understanding of the dynamics of expressions. Another goal is to discover which properties of the face should be simulated/rendered with the most detail. For instance, is skin blushing or muscle bulging as important as wrinkles?

Acknowledgments

The authors would like to express their appreciation to Keith Waters for his willingness to share his facial data and deformation equations, as well as helpful EMail discussions. We would also like to thank Carol Wang of the University of Calgary for forwarding example code based upon Dr. Waters' early work, and for many interesting discussions.

References

[Ekman and Friesen 77] P. Ekman and W. V. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, 1977.

[Terzopoulos *et al.* 88] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, Vol. 36, pp.91–123, 1988.

[Wang 92] C. Wang. Personal communication, 1992.

[Waters 87] K. Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics*, Vol. 22, No. 4, pp.17–24, 1987.