From Real Faces To Virtual Faces: Problems and Solutions

WON-SOOK LEE, NADIA MAGNENAT THALMANN

MIRALab, CUI, University of Geneva, Geneva, Switzerland
E-mail : {wslee, thalmann}@cui.unige.ch

Today, a lot of research is going on the automatic cloning of real humans to be able to see the other 3D person in real time in the virtual worlds. Using networked virtual environment systems, we are able to converse in a virtual room. To do this, we have first to clone 3D heads and emotions to be able to simulate them. This problem is very difficult to solve for several reasons that this paper will analyze. Also the results obtained should be realistic looking and believable. In this paper, we analyze our experience to generate a virtual face for animation according to given inputs and compare their drawbacks and efficiency. We also introduce our animation system.

Keywords: reconstructions, feature points, generic model, deformation, texture mapping, range data, picture data.

1 Introduction

To clone a face has attracted people for a long time, in the real world and virtual world. With the growing power of computer speed and multimedia ability, people would like to have their counterpart in virtual world and animate it and utilize it for communication in an efficient way. However until now to get a realistic facial reconstruction in commercial equipment and very detailed matched face has been considered as a growing research area. We show our main three methods to generate a clone from a given input; such as one or several random pictures of a person, orthogonal pair of front and side views of a person or range data obtained with more sophisticated equipment or a rather complicated algorithm. Table 1 classifies these three inputs and results.

Input	Equipment	Method	Result	Time
Pictures (unorganized)	Mono camera	Manual using user friendly designing software.	Detailed matching with human eyes	days/ weeks
Pictures (organized)	Mono camera	Automatic methods using Feature detection and generic model modification	Rough matching	minutes
Range data (detailed shape, no animation structure)	Laser scanner/ Light stripper/ Stereoscopic camera + extra	Automatic methods using Feature detection and generic model modification and fine adaptation.	Detailed matching	minutes

Table 1 Three possible ways to get a cloned face for animation in virtual world.

2 Unorganized picture data

There are many ways to make a clone of a person who is not available to have pictures with special purpose for instance Marylin Monroe. Sometimes only one picture is available, or several pictures taken at different times and places which have great difficulty to get 3D data without help from human eyes. To get a detailed shape, we need to do a time-consuming manual job with help of user friendly graphical interface.

Plaster Model Magnenat Thalmann et al. [15] used plaster models in real world and selected facets and vertices marking on the models which are photographed from various angles to be digitized. Then all the 2D coordinates are combined to produce 3D data of the face. Here the reconstruction approach requires a mesh drawn on the face and is time consuming, but can obtain high resolution in any interested area. Pixar's new animation character "Geri"[1] is sculpted, which was then digitized and used as a basis for creating the 3D model.

Interactive Deformation With one or several pictures, we use our *Scupltor* software [18], dedicated to the modeling of 3D objects. The sculpting approach is based on local and global geometric deformations. Adding, deleting, modifying, assembling triangle meshes are the basic features provided by *Sculptor*. Real-time deformations and manipulation of the surface gives the designers the same facilities as with

real clay or wax sculpting. There are two ways to create a face. One way to start from a template head, this therefore accelerates the creation process. The second method from scratch, the designer can model half of the head and use a symmetric copy for the other half. At the end, small changes should be made on the whole head because asymmetric faces look more realistic.

Figure 2-1 Head creation from a template in *Sculptor* and *TextureFit* program. The upper left image is a photo of the real terra-cotta soldier. The upper middle image is the texture created from the photo. The other objects are snapshots from the 3D model.

Interactive Texture Mapping Texture mapping is a well-known low-cost method in computer graphics for improving the quality of virtual objects by applying real images onto them. For virtual humans, the texture can add a grain to the skin, including the color details like color variation for the hair and mouth. These features require correlation between the image and the 3D object. A simple projection is not always sufficient to realize this correlation: the object, which is designed by hand, can be slightly different from the real image. Therefore an interactive fitting of the texture is required. A program allowing the fitting of the texture according to features of the 3D object is called *TextureFit* [2]. This enables the designer to interactively select a few 3D points on the object. These 3D points are then projected onto the 2D image. The projection can be chosen and set interactively, hence the designer is able to adjust these projected points to their correct position on the image. Figure 2-1 shows the input and resulted head produced using *Scupltor* and *TextureFit.*

Animation Structure To simulate the effects of muscle actions on the skin of

virtual human face, specific regions are defined on the mesh corresponding to the anatomical regions where a muscle is desired. A control lattice is then defined on the region of interest while modeling is processed. This method is necessary when a person is not available at that specific time & place, and specially when we want to generate a character in imagination. The result depends on a designer's artistic sense and usually it required at least days and weeks time. To give an animation structure is also another deal since every generated character has different points and faces structures.

3 Organized Picture data

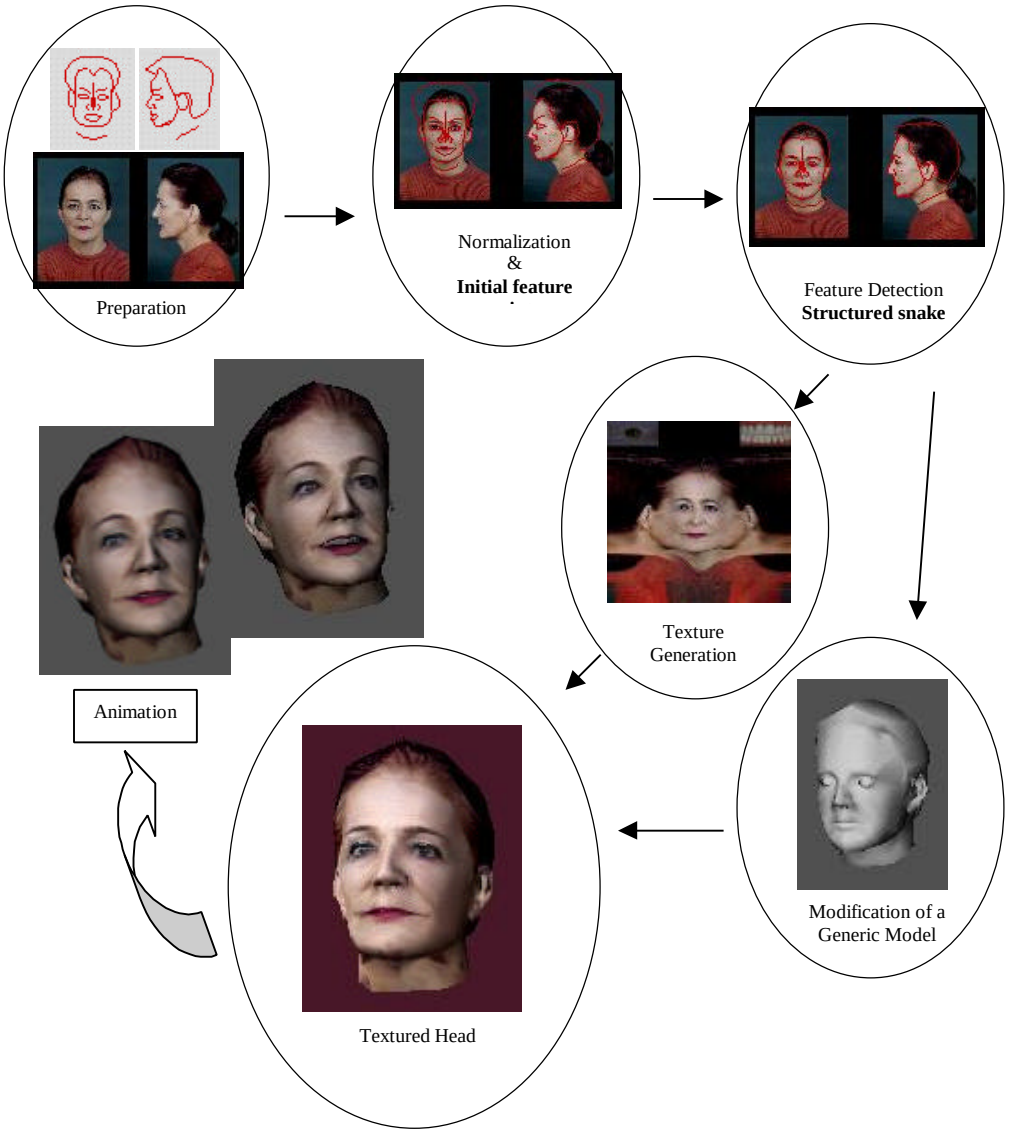

Figure 3-1 Overall flow for 3D-head reconstruction from two orthogonal pictures.

There are faster approaches to reconstruct a face shape from few pictures of a face. In this method, a generic model with animation structure in 3D is provided in advance, and a limited number of feature points, which are the most characteristic points to recognize people, detected either automatically or interactively on the two (or more) orthogonal pictures, and

the other points on the generic model are modified by a special function. Then 3D points are calculated by just combining several 2D coordinates.

Kurihara and Arai [12], Akimoto et al. [5], and Ip and Yin [9] use an interactive method or automatic method to detect feature points and modify a generic model. Each of them has some drawbacks such as too few points to guarantee appropriate shape from a very different generic head or accurate texture fitting, or too much loose automatic methods like simple filtering and texture image generation using simple linear interpolation blending. We present our approach to reconstruct a real face from two orthogonal views. In this reconstruction, feature points are extracted from front and side views. Reconstruction of a shape may not require high accuracy in some special cases, thanks to the texture mapping. However, for animation, we need a process of texture fitting which can ensure positional correspondence of features in the model and the texture image. The reliability of texture fitting is based on the number of feature points. To get better result we need more points at the right place. If an automatic method is not robust enough, it is better to use interactive way. See Figure 3-1.

3.1 Preparation & Normalization

First we prepare two 2D wire frames composed of feature points with predefined relation for front and side views. The frames are designed to be used as an initial position for the snake method later on. Then we take pictures from front and side views of a head. The picture is taken with maximum resolution and the face is in neutral expression and pose.

To make the head heights of side and front views the same, we measure heights of them, and choose one point from each view for matching them with corresponding points in prepared frame. As an example we select the highest point on a front face and the top of the nose on a side face. Then we use transformation (scaling and translation) to bring the pictures to the wire frame coordinate, overlaying frames on pictures.

3.2 Feature Detection

We provide automatic feature points extraction method with an interface for interactive correction if and when needed. We consider hair outline and face outline and some interior points such as eyes, nose, lips, eyebrows and ears as feature points. There are methods to detect them just using special background information and predefined threshold [5][9] and then use an edge detection method and apply threshold again. Also image segmentation by clustering method is used [5]. However, it is not very reliable since the boundary between hair and face and chin lines are not easy to detect in many cases. Moreover color thresholding is too sensitive depending on each individual's face image and therefore requires many trials and experiments. We use a structured snake, which has functionality to keep the structure of contours. It does not depend on the background color much and is more robust than simple thresholding method.

3.2.1 Structured Snake

First developed by Kass et al. [11] active contour method, so called as snakes, is widely used to fit a contour on a given image. This allows the fitting of the boundary points of maximum contrast close to the already-defined rough contour. To get correspondence between points from pictures and points on a generic model which has a defined number, a snake is a good candidate. Above the conventional snake, we add three more functions. First, we move few points to the corresponding position interactively, and anchor them. It helps later to keep the structure of points when snakes are involved and is also useful to get more reliable result when the edge we would like to detect is not very strong. We then use color blending first for a special area, so that it can be attracted by a special color [6]. When the color is not very helpful and Sobel operator is not enough to get good edge detection, we use a multiresolution technique [7]. We can insert some more points which are not visible, but work as a member of snake to keep non uniform interval between visible points of the snake.

We have several parameters such as elasticity, rigidity, image potential and time step, to manage the movement of snake. As our contours are modeled as polylines, we use a discrete snake model with elastic, rigid forces and image force acting on each pixel for color interest. We define different sets of parameters for hair and face according to their color characteristic. To adjust the snake on points of strong contrast, we consider

$$F_{ext,\,i} = n_i \cdot \tilde{N} E(v_i) \qquad (1)$$

where n_i is the normal to the curve at the node i, whose position is v_i and is given by

$$E(v_i) = | \tilde{N} I(v_i) |^2 \qquad (2)$$

where $I(v_i)$ represents the image itself. To estimate the gradient of the image, we use the Sobel operator. Blending of the different color channels is changed to alter the snake's color channel sensitivity. For instance, we can make snakes sensitive to the excess of dark brown color for hair. Also we use clamping function to emphasize special interesting range of color. Snakes are useful in many circumstances, particularly in the presence of high contrast. The color blending and clamping function depend on the individual.

Sobel operator does not always provide strong edge for some areas , for instance, chin lines. In such cases, we employ multiresolution approach [7] to obtain strong edges. It has two main operators, REDUCE with Gaussian operator and EXPAND. The subtraction produces an image resembling the result after Laplacian operators commonly used in the image processing. More times the REDUCE operator is applied stronger are the edges.

3.3 Modifying a Generic Model

We produce 3D points from two 2D points on frames with predefined relation between points on a front view and on a side view. Some points have x, y_f, y_s, z, so we take y_s , y_f or average of y_s and y_f for y coordinate (subscripts s and f mean side and front view). Some others have only x, y_f and others x, y_s. Using predefined relation from a typical face, we get 3D position (x, y, z). We modify non-feature points with some distance-related functions, spring-mass model, or FFD method. Here we employ Dirichlet Free Form Deformations (DFFD) to move other points according to feature points.

3.3.1 DFFD

Distance-related functions have been employed by many researchers [5][9][12] to calculate displacement of non-feature points related to feature points detected. We propose to use DFFD [14] since it has capacity for non-linear deformations as opposed to generally applied linear interpolation which can give smooth result for the surface. The process of modification of a generic model fitting feature points detected from given two orthogonal pictures are as follows.

1. We define control points on a generic model which are corresponding to feature points detected from two views of a person and 27 points for a box surrounding.

2. We apply global transformations (translation, and scaling) to bring detected feature points to generic model's 3D space. We compare two eye extremities in a generic model and a specific person for scaling. We check the center of rightmost, leftmost, up-most, down-most, front-most, and back-most points of the head for translation.

3. We apply the DFFD on the points of the generic head. The displacement of non-feature points depends on the distance between control points. Since DFFD applies Voronoi and Delaunay triangulation, some points outside triangles of control points are not modified, the out-box of 27 points can be adjusted locally.

4. Eyes and teeth are recovered to the original shape since modifications may create unexpected deformation for them. Here translation is used after comparison of a generic and individualized heads.

Our system provides a feedback modification of a head between feature detection and a resulted head.

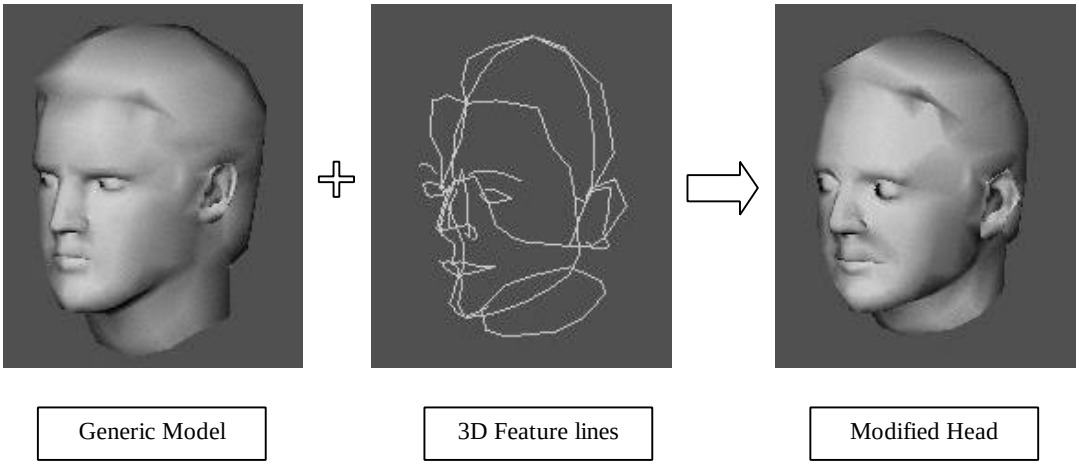

Figure 3-2 A result of DFFD modification comparing with the original head.

3.4 Automatic Texture Mapping

To increase realism, we utilize texture mapping. Texture mapping needs a texture image and coordinate for each point on a head. Since our input is two pictures, texture image generation is needed.

3.4.1 Texture Image Generation

For smooth texture mapping, we assemble two images from front and side views to be one.

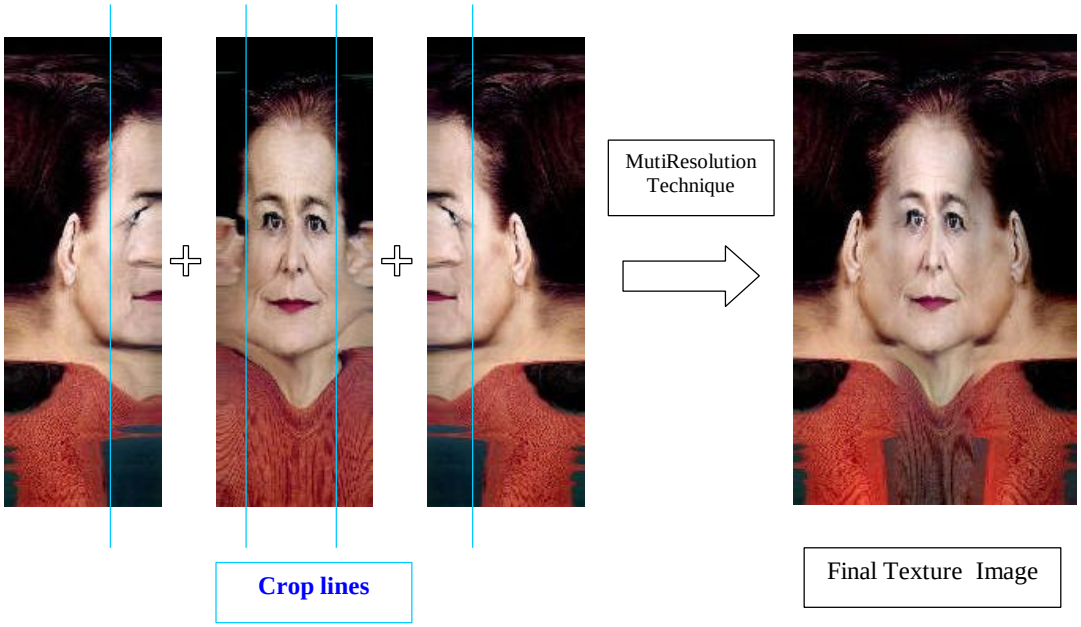

Figure 3-3 Texture Generation from front and side view images. Front and right view after cylindrical projection covering 180^0 for each. We crop the front and side views around eye ends (shown through blue lines) and combine with the left view (flipped from the right view). The last image shows image mosaic of three images, front, right and left using multiresolution spline method.

1. boundaries of two pictures are detected using boundary color information. Since the hair shape is simple in a generic model, the boundary of a side view is modified automatically using information of back head profile feature points detected. It is useful to have nice texture for back part of a neck.
2. The cylindrical projection of each image is done. So the front view will cover from -90^0 to 90^0, right view from 0^0 to 180^0 and left one from -180^0 to 0^0.
3. Cropping of two images are processed. Since the front view is good only for a certain range and so is the other. To use conventional blending for wide range using angle variation [9], it is easy to blur certain shape of features. Images for certain points (we use eye extremes because eyes are important to keep high resolution) are cropped on front and side views. Since we have information about eye positions, it can be done automatically to crop a front view, then we crop right and left views to make the final assembled image be 360^0.
4. Image mosaic is applied to produce one image for texture mapping. Since it is almost impossible to take two orthogonal pictures in exactly same condition, just to assemble several images makes the boundaries visible. We use a multiresolution spline method to assemble two images [7].

The cylindrical projections of front and right views and the image mosaic using

a multiresolution spline are shown in Figure 3-3.

3.4.2 Texture Fitting

Texture fitting with a composed image is employed to 3D modified head. The main idea for the texture fitting is to map a 2D image to a 3D shape. Texture coordinates of feature points are calculated using detected position data and a function and cropping lines applied for texture image generation. The problem for texture fitting is how to decide texture coordinates of all points on a surface of a head. The process is as follows.

1. Cylindrical projection to 2D is applied for all points on a surface.
2. Extra points are added to make a convex hull containing all points. Extra points are designed to have texture coordinates on the texture image.
3. The Voronoi triangulation on control (feature) points and extra points is processed.
4. Local Barycentric coordinates of every point with a surrounding Voronoi triangle is calculated.
5. Texture coordinates of each point on a 2D-texture image is obtained using texture coordinates of control points and extra points and corresponding Barycentric coordinate.

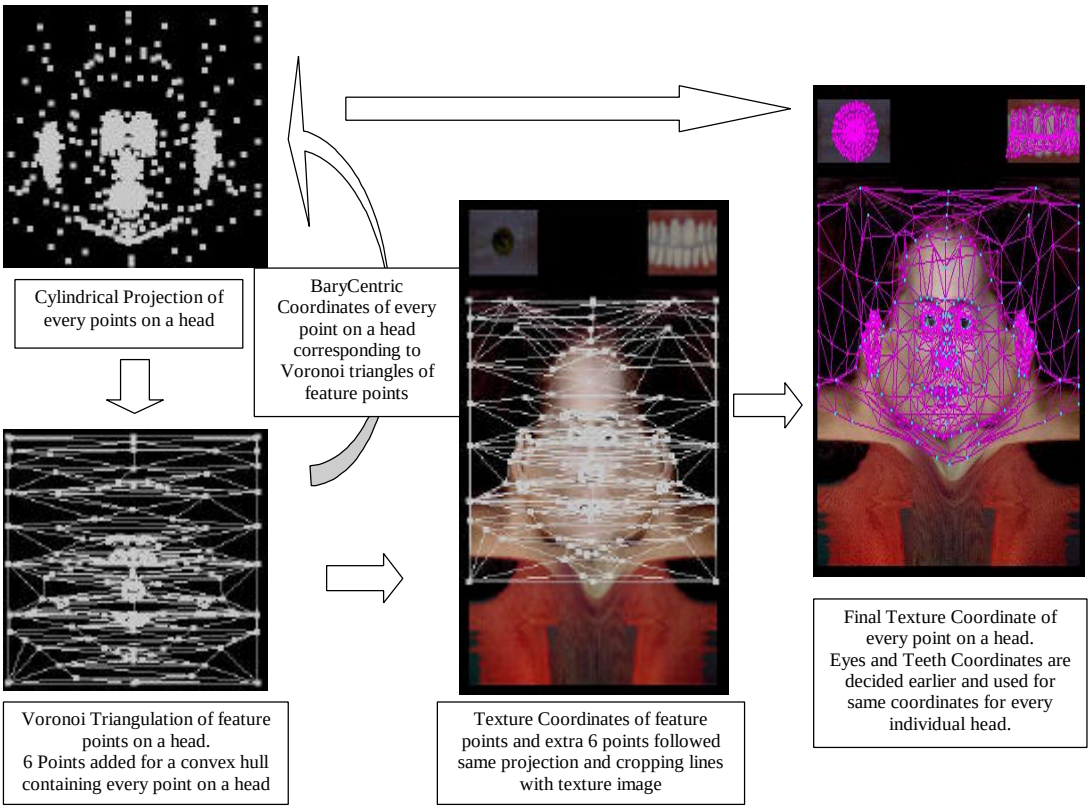

Figure 3-4 Texture fitting process.

3.5 Result and Comparison

A final textured head is shown in Figure 3-6 with input in Figure 3-5, whose process from feature detection to texture mapping takes a few minutes. Compare the output with the head in Figure 2-1 produced by a designer using *Sculptor* and *TextureFit* which takes weeks of efforts. It has a

smoother texture image since it is improved by manual correction by a designer.

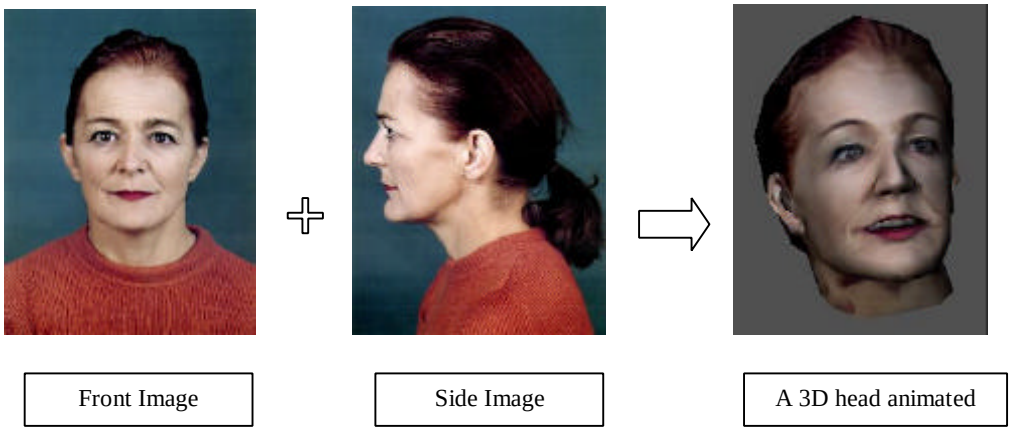

Figure 3-5 Input pictures and final animated head

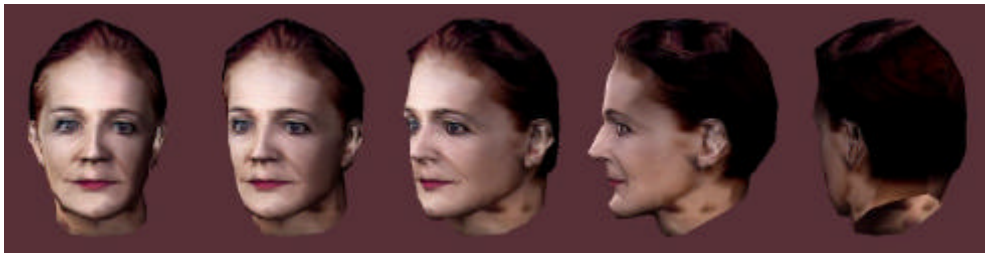

Figure 3-6 A final reconstructed head. A backside has proper texture too.

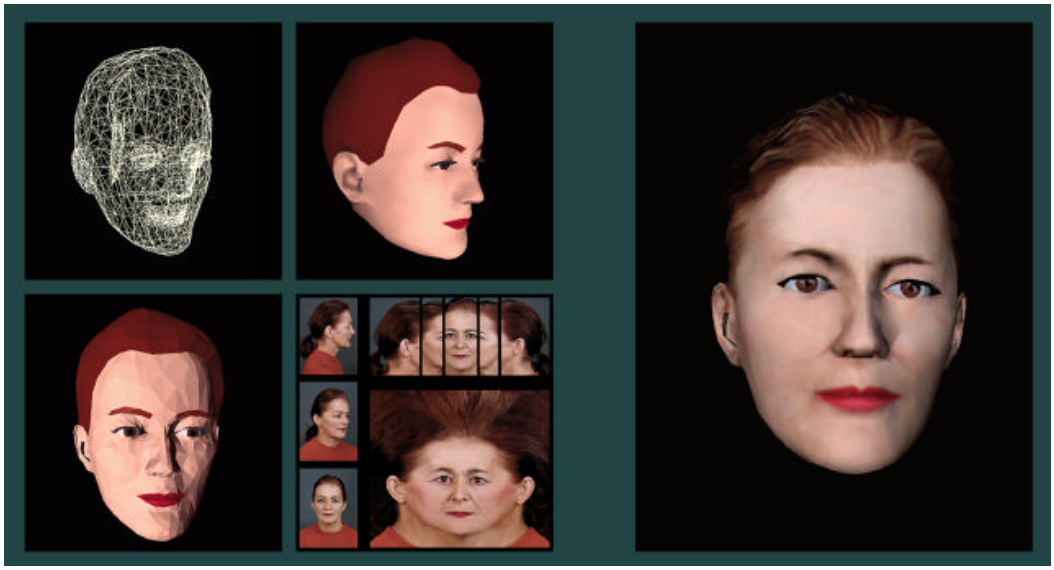

Figure 3-7 Face reconstruction by a designer with three pictures (front, diagonal, and side views) using *Sculptor* and *TextureFit* mentioned in Section 1. A texture image part is composed using *PhotoShop*.

4 Range Data

The approach based on 3D digitization to get a range data often requires special purpose high-cost hardware. However when we want to get really highly matched face, it is necessary to try to have a range data. These data provide with a large number of points

usually and it does not have any structure for animation. It is the same as a paper in topological view.

4.1 How to get Range Data?

We describe the main three methods to get range data.

Laser Scanning In range image vision system some sensors, such as scanners, yield range images. For each pixel of the image, the range to the visible surface of the objects in the scene is known. Therefore, spatial location is determined for a large number of points on this surface. An example of commercial 3D digitizer based on laser-light scanning, is Cyberware Color Digitizer™ [17]. Lee et al. [13] digitized facial geometry through the use of scanning range sensors. However, the approach based on 3D digitization requires special high-cost hardware and a powerful workstation.

Stripe Generator As an example of structured light camera range digitizer, a light striper with a camera and stripe pattern generator can be used for face reconstruction with relatively cheap equipment compared to laser scanners. Stripe pattern is projected on the 3D object surface and it is taken by a camera. With information of positions of projector and camera and stripe pattern, a 3D shape can be calculated. Proesmans et al. [16] shows a good dynamic 3D shape using a slide projector, by a frame-by-frame reconstruction of a video.

Stereoscopy A distance measurement method such as stereo can establish the correspondence at certain characteristic points. The method uses the geometric relation over stereo images to recover the surface depth. C3D 2020 capture system by the Turing Institute produces many VRML models using stereoscopy method [3].

Most methods have great problem to get hair shape because of the high reflection structure.

4.2 How to give structure?

To get a structured shape for animation, most typical way is to modify an available generic model with structural information such that eyes, lips, nose, hair and so on. Our input is any VRML format with texture image, point coordinate data and texture coordinate data. For our experiment, we utilized data copied from the Turing Institute [4] where stereoscopic camera is used to generate range data of faces. It has only front face image. We apply similar method with the one in Section 3 with an adaptation method utilizing available detailed shape data. We utlilze two step adaptation method**,** which has rough matching and then detailed matching later. See Figure 4-1 for overall process.

More detailed process follows.

1. Feature detection for front face picture is processed. Whenever a feature point is detected, the corresponding depth, *z* coordinate, is obtained automatically calculating its neighboring points of given *x* and *y* position. Since most cases, a feature point does not corresponding to any point in the range data, we collect certain number of nearest points in terms of *x* and *y* and then we calculate reverse distance function ratio to get the depth.
2. Some points need to be corrected interactively since the hair part is not detected well in the range data

aquirement. Also back head shape needs to be imagined.

3. DFFD is applied with ontained 3D coordinates of given feature points. The result is a rough matching.
4. Feature points which are gained from range data are collected using Voronoi triangulation and Barycentric coodinate calculation. The Voronoi triangles and detected points on a surface are shown in Figure 4-2.
5. Various projections of points in right picture in Figure 4-2 are used with relation to a normal vector of a corresponding triangle. Once again the nearest points are calculated and reverse distance function is applied to get the corresponding accurate coordinate in a range data.

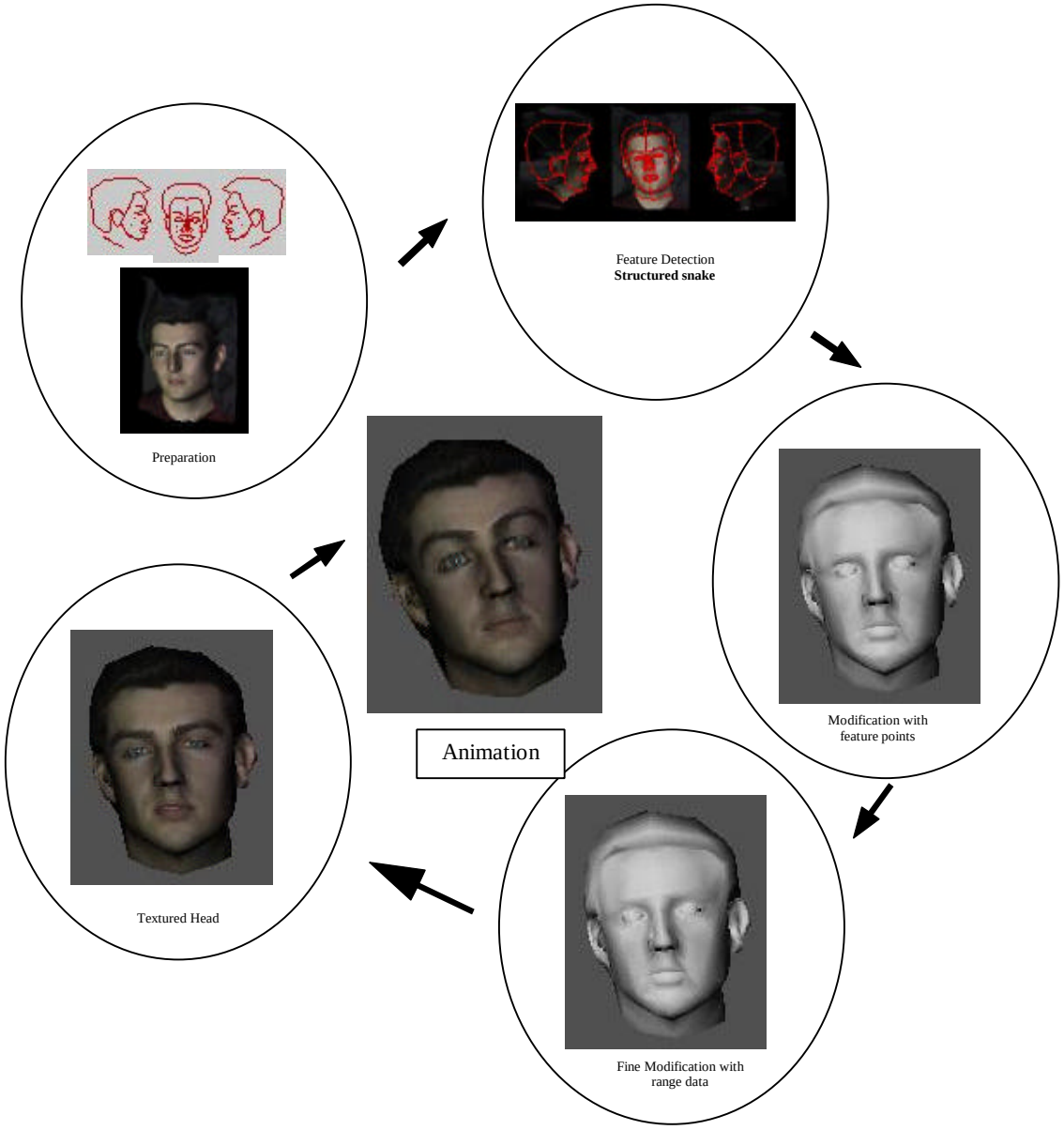

Figure 4-1 The overall process for giving animation structure for range data. The input range data is copied from the Turing Institute. This process can be applied to any range data in VRML format.

Without the first modification step with feature points, it is difficult to apply projection of points to get detailed matching.

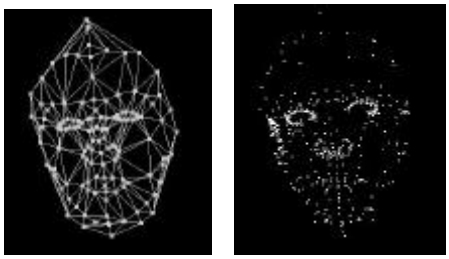

Figure 4-2 Left picture shows Voronoi triangles of feature points which are used for fine modification and the right points which fall inside region for fine modification.

5 Facial Animation System

We employ an approach for deformation and animation of a face, based on pseudo muscle design. The (generic) face model is an irregular structure defined as a polygonal mesh. The face is decomposed into regions where muscular activity is simulated using Rational Free Form Deformations [10]. As model fitting transforms the generic face without changing the underlying structure, the resulting new face can be animated. To simulate the effects of muscle actions on the skin of virtual human face, we define regions on the mesh corresponding to the anatomical descriptions of the regions where a muscle is desired. For example, regions are defined for eyebrows, cheeks, mouth, jaw, eyes, etc. A control lattice is then defined on the region of interest. Muscle actions to stretch, expand, and compress the inside geometry of face are simulated by displacing or changing the weight of the control points. This deformation model for simulating muscle is simple and easy to perform, natural and intuitive to apply and efficient to use for real time applications.

Facial Motion Control Specification and animation of facial animation muscle actions may be tedious task. There is a definite need for higher level specification which would avoid setting up the parameters involved for muscular actions when producing an animation sequence. The Facial Action Coding System (FACS) [10] has been used extensively to provide a higher level specification when generating facial expressions, particularly in nonverbal communication context. In our multi-level approach as shown in Figure 5-1, motion parameters as Minimum Perceptible Action (MPA) has a corresponding set of visible features such as movement of eyebrows, jaw, or mouth and others occurring as a result of muscle contractions and pulls. The MPAs are used for defining both the facial expressions and the visemes. There are 65 MPAs used in our system which allow us to construct practically any expression and viseme. At the highest level, animation is controlled by a script containing speech and emotions with their duration. Depending on the type of application and input different levels of animation control can be utilized.

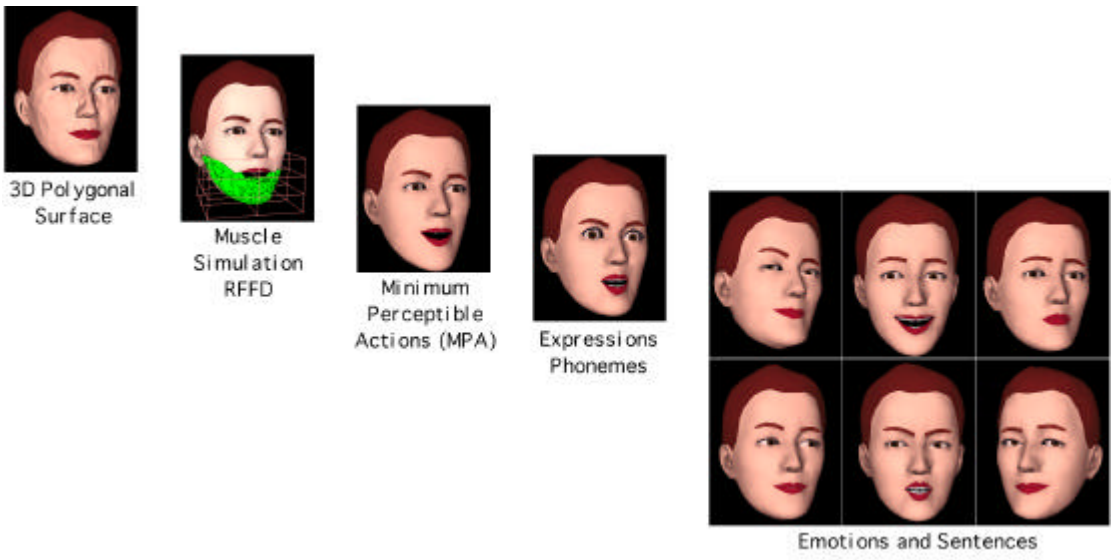

Figure 5-1 Different levels of facial motion control.

6 Conclusion

We described our experience to create a virtual animated face from a real face and compare their inputs and results. First designer oriented reconstruction has a space for artistic sense, but it is time consuming. The second reconstruction method modifying a generic model using two orthogonal pictures needs commercial equipment and takes just few minutes. The third method using range data has the best visual result for output and time, but it requires usually either an expensive or sophisticated equipment.

To have a counter part of a real face into virtual world has a lot of potential in the fields from entertainment to medical application, such as 3D morphing, simulation of generating population, face to face communication through network, and face surgery simulation.

The integration of our reconstruction method of a 3D head with facial feature tracking from video sequence is our ongoing research topic.

7 Reference

[1] Meet Geri: The New Face of Animation, Computer Graphics World, Volume 21, Number 2, February 1998.

[2] Sannier G., Magnenat Thalmann N., "A User-Friendly Texture-Fitting Methodology for Virtual Humans", Computer Graphics International'97, 1997.

[3] Exhibition On the 10th and 11th September 1996 at the Industrial Exhibition of the British Machine Vision Conference.

[4] http://www.turing.gla.ac.uk/turing/copyrigh.htm

[5] Takaaki Akimoto, Yasuhito Suenaga, and Richard S. Wallace, Automatic Creation of 3D Facial Models, *IEEE Computer Graphics & Applications*, Sep., 1993.

[6] P. Beylot, P. Gingins, P. Kalra, N. Magnenat Thalmann, W. Maurel, D. Thalmann, and F. Fasel, 3D Interactive Topological Modeling using Visible Human Dataset.

Computer Graphics Forum, 15(3):33-44, 1996.

[7] Peter J. Burt and Edward H. Andelson, A Multiresolution Spline With Application to Image Mosaics, *ACM Transactions on Graphics*, 2(4):217-236, Oct., 1983.

[8] Marc Escher, N. Magnenat-Thalmann, Automatic 3D Cloning and Real-Time Animation of a Human Face, *Proc. Computer Animation*, IEEE Computer Society, pp. 58-66, 1997.

[9] Horace H.S. Ip, Lijin Yin, Constructing a 3D individual head model from two orthogonal views. *The Visual Computer*, Springer-Verlag, 12:254-266, 1996.

[10] Kalra P, Mangili A, Magnenat-Thalmann N, Thalmann D, Simulation of Muscle Actions using Rational Free Form Deformations, Proc Eurographics '92, Computer Graphics Forum, Vol. 2, No. 3, pp. 59-69, 1992.

[11] M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models, *International Journal of Computer Vision*, pp. 321-331, 1988.

[12] Tsuneya Kurihara and Kiyoshi Arai, A Transformation Method for Modeling and Animation of the Human Face from Photographs, *Computer Animation*, Springer-Verlag Tokyo, pp. 45-58, 1991.

[13] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters, Realistic Modeling for Facial Animation, In *Computer Graphics (Proc. SIGGRAPH)*, pp. 55-62, 1996.

[14] L. Moccozet, N. Magnenat-Thalmann, Dirichlet Free-Form Deformations and their Application to Hand Simulation, *Proc. Computer Animation*, IEEE Computer Society, pp. 93-102, 1997.

[15] N. Magnenat-Thalmann, D. Thalmann, The direction of Synthetic Actors in the film Rendez-vous à Montrèal, *IEEE Computer Graphics and Applications*, 7(12):9-19, 1987.

[16] Marc Proesmans, Luc Van Gool. Reading between the lines - a method for extracting dynamic 3D with texture. In *Proceedings of VRST*, pp. 95-102, 1997.

[17] http://www.viewpoint/com/free stuff/cyberscan

[18] LeBlanc. A., Kalra, P., Magnenat-Thalmann, N. and Thalmann, D. Sculpting with the 'Ball & Mouse' Metaphor, Proc. Graphics Interface '91. Calgary, Canada, pp. 152-9, 1991.