Physics based facial modeling and animation

by

LEUNG Hoi-Chau

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Automation and Computer-Aided Engineering

The Chinese University of Hong Kong
September, 2002

Abstract

This thesis presents a technique for real time physics based facial modeling and animation. The proposed technique for the deformation of a linear elastic facial model is based on the Boundary Element Method (BEM). By using facial modeling, facial feature recognition and facial muscle recognition processes in the system, different facial expressions can be efficiently produced.

The system starts from acquiring a facial model represented by mesh. The facial model will then be offset to a closed surface for the BEM engine to manipulate. Essential facial features of the facial model are automatically recognized by the system. A number of facial muscles are then located based on the positions of the facial features. Muscle based controller is used to deform the facial model in generating different facial expressions.

Solving the system of equations governing the facial expression deformation in every muscle contraction and relaxation is not practical and cannot achieve a good frame rate. The pre-computation and the matrix inversion update techniques are adopted to speed up the calculations.

Finally, we have demonstrated the performance and usefulness of the system are illustrated with a number of facial models.

摘要

本論文提供一建於物理基礎上和實時的面部造型及動畫技術。我們利用邊界元素方法〔Boundary Element Method〕來限定線性彈性面部模型的變形。透過面部造型、面部特徵識別及面部肌肉識別等運算程序，系統可有效率地造出不同的面部表情。

首先，系統須取得一網格面部模型。然後，系統會基於已輸入的面部模型，以偏移方法得出一閉面模型，給 BEM 處理器用。系統會自動識別面部模型上重要的面部特徵，從而確定一系列面部肌肉在其模型上的位置。由此建立出來的面部肌肉控制器，能定義面部模型的變形方式，以造出不同的面部表情。

我們知道如每當肌肉收縮或鬆弛時都尋求其相應之方程解並不實際，以及難以合乎令人滿意的幅率。此系統利用預計算和矩陣逆算更新技術以加快運算速度，令系統實時地得出不同的面部表情。

最後，本論文更以多個不同的面部模型測試本系統，以論證其性能和效用。

Contents

Chapter 1. Introduction

Facial expressions take a major role in character modeling and animation. Traditional facial expressions are mostly generated by blending different morph targets into a final shape. Usually, a large number of morph targets are used and they are modeled manually.

Our physics and muscle based facial modeling technique enable the automation of morph target modeling or a substitution of the blending technique. In our system only one front face mesh is required. All other parameters required by the calculation can be automatically generated by the system. Different facial expressions can be produced by deforming the facial muscles on the original facial model. By following the true facial muscle profile, we can simulate most of the facial expressions effectively on a virtual human face.

Chapter 2. Previous Works

There are two main streams in facial modeling: they are the facial animation technique which mainly focuses on the speed of the system; and the facial surgery simulation which focuses on realism. Physics based modeling technique such as Finite Element Method (FEM) is used extensively in surgery simulation. It seems that both realism and interactivity are conflicting each other in these systems. More discussions on these systems will be given later in this chapter.

2.1 Facial animations and facial surgery simulations

In this section various techniques and works on facial modeling will be discussed. During these few decades, several techniques and applications have been explored for facial modeling. They range from simple parametric local deformation to physics based numerical methods (e.g. FEM); applications from facial animation to facial surgery simulation. Most of the time, facial animation relies on the simple techniques such as parametric and spring-mass deformation, while facial surgery simulation relies on the FEM.

For most facial animation systems, the facial expressions are reproduced by controlling the muscles embedded in the facial tissues. By changing the length or shape of these muscles, the facial mesh deform accordingly.

In order to deform the facial mesh of a character or an individual, we must have the mesh first. Among all those systems, there are mainly three different sources to obtain the facial mesh, they are i) Range data from 3D scanner, ii) Iso-surface extraction from volume data such as CT scan and MRI, and iii) CAD model.

In 1981, Stephen *et al.* [1] described a facial modeling system which takes an image as input and outputs a facial animation. The system performs the image processing, feature recognition and constructs the FACS [19] representations of the facial model and then applies forces to deform the facial mesh. However only the FACS to facial animation parts have been implemented. Since then many other researches on facial animation [3] [17] have used FACS to test their systems and assign expressions to their facial models.

Another attempt to automate the facial animation process is made by Yuencheng *et al.* [7] [9]. They make use of image processing techniques to adapt any individual facial mesh acquired by a Cyberware scanner to a generic facial mesh. Three layers spring-mass models are used to simulate the skull, fatty tissue and skin. Forces are applied to the facial model through muscles which attached to the skin tissue and the facial skeleton. Facial animation is then obtained by solving a system of equations governing the motions of the model.

Keith [3] used a muscle vector to deform the neighbors of the facial nodes. The resulting facial mesh is of C^0 continuously.

Jean-Paul *et al* [4] proposed a technique for simulating skin deformations in a grasping task. They use FEM to simulate the interaction between the hand and an object when they are in contact; and Joint-dependent Local deformation (JLD) to deform the hand in non-contact task. The finite elements are formed between the mesh of the hand and the bones.

George *et al.* [5] uses shell elements in FEM to simulate free form sculpting. The deformable model's shape is calculated indirectly by finding a minimum to an energy functional or by solving a set of differential equations.

A biomechanical-based muscle is modeled by David *et al.* [6]. To test the biomechanical validity of the muscle model developed, two well-known experiments are simulated and compared to the results in the literature. All their simulations are based on the FEM.

A craniofacial surgery simulation is carried out by H. Delingette *et al.* [8]. Their system makes use of the simplex-mesh [20] extensively which is developed by H. Delingette.

To speed up the surgery simulation for the FEM engine, Morten *et al.* [10] make uses of the Condensation and the Selective Matrix Vector Multiplication (SMVM) techniques.

An anatomy-based facial surgery planning system is implemented by Erwin *et al.* [11]. The system extracts the skull from the computer tomography volume data set and the facial surface mesh from the laser scan of a patient. A solid element mesh is then constructed. Operations such as cutting and realignment of the skull are allowed. The postoperative appearance of the facial mesh is computed by the FEM.

R. M. Koch *et al.* [12] developed a facial surgery simulation system based on FEM shell elements and elastic springs. Surface mesh of the face and skull are extracted from the Visible Human Data Set™ (VHD) [21]. Whereas using the same stretching and blending parameters over the entire facial mesh, different values are assigned to different parts of the facial surface mesh. The springs connecting the surface mesh of the face and the skull are divided into several shorter springs with different stiffness. Cutting and realignment operations are preformed on the skull, the FEM engine finds the deformed facial mesh after solving a system of equations.

Jane *et al.* [14] used the anatomically based approach to model and animate animals. Given a skeletal model and the muscle location, the system generates the

iso-surface of the animal and then anchors the skin vertices to the underlying structure. In this way deformation of the muscles causes the deformation of the animal skin. The whole anatomy structure is linked by a spring-mass system.

Ferdi *et al.* [13] also focused on the anatomically based modeling. In their system the tendon and insertion of the muscles are considered. Muscles are implemented as ellipsoids and skins are the implicit form of the overall muscles.

S. H. Martin *et al.* [16] used Berstein-Bézier functions as the shape functions for their FEM engine.

Rolf *et al.* [17] extended their previous system [12] to the facial surgery simulation level. The major work of their new system is the pre-computation of the Facial Actions of the FACS, superposition is then used to combine several facial actions to express an emotion.

Qing-hong *et al.* [15] used voxel and FEM techniques to model muscle deformation. In their system the muscle is represented by the voxels which are also the elements for the FEM engine. The deformed muscle is then sampled into voxels afterwards.

Haptic feedback device is included in Stéphance *et al.* [18] surgery simulation system. Users manipulate the haptic feedback device to deform the object which is extracted from the medical volume data set. The deformation is based on the FEM and a quasi-nonlinear component is added to simulate nonlinear visco-elasticity. In order to allow real-time deformation, a pre-process is applied before evaluating deformation. A superposition technique is then used to calculate the deformation at run-time.

2.2 Facial Action Coding System (FACS)

The Facial Action Coding System aims at providing a comprehensive classification of all possible visually distinguishable facial movements. A single action unit describes one visually distinguishable facial movement which can be produced by two or more different muscles. Therefore, an emotion can be expressed by simply combining different action units together.

2.3 The Boundary Element Method (BEM) in Computer Graphics

The first paper which introduces the Boundary Element Method to the computer graphics and CAD area is from Doug L. James and Dinesh K. Pai [22]. Their ARTDEFO system makes use of the BEM technique to deform a virtual object by interacting with a virtual hand. In 2002, K. C. Hui and H. C. Leung [23] applied the BEM technique to a volumetric sculpting system.

BEM only requires the surface mesh of a solid object to perform the linear elastic deformation. This gives its advantage over the FEM in computer graphics.

Chapter 3. The Facial Expression System

The Facial Expression System combines the boundary element method (BEM) and some facial feature recognition technique to give a facial muscle and physics based animation system. The system takes as input a shell liked triangulated polygonal mesh to generate all other data required. The system is composed of a facial modeling module which provides a valid input to the BEM engine; a facial feature recognition module, and the module for locating facial muscles.

The BEM engine makes use of the physics based numerical stress analysis technique "Boundary Element Method" to perform the calculations of a linear elasticity deformation for the facial mesh. When the locations of the facial muscles are determined, user can use the data generated by the system to initialize the BEM engine. Once the BEM engine finished the computation, ten facial muscle sliders are provided to control the facial muscles in the system, and thus different facial expression can be made. Since the computation time for the initialization of the BEM engine can be quite long, the data generated from the BEM initialization process can be saved for later use.

The following sections describe the input to the system, the facial modeling and feature recognition tasks, the user control and the output of the system.

3.1 Input to the system

The input mesh consists of features that are common to the human face as discussed below.

3.1.1 Orientation requirements for the input mesh

The input facial mesh is required to be of a specified orientation. The front face is in the x-y plane of the system, with the eyes in the positive y direction of the mouth and the left eye in the positive x direction of the right eye.

3.1.2 Topology requirements for the input mesh

The topology of the facial mesh affects the facial expression that can be produced by the system. The most obvious one is the opening of the mouth; if the regions around the mouth on the facial mesh are closed, there is no way for the system to produce a facial expression that is with its mouth opened. Therefore, there should be a hole in the regions around the mouth in the input mesh. This hole gives the *Mouth Contour* in Figure 3.1. For the same reason in the modeling of human face, there are always two holes in the regions around the left and right eye. These two holes are expected in the input mesh and they are the *Left Eye Contour* and *Right Eye Contour* in Figure 3.1. Since most of the facial expressions are produced by controlling the facial muscles in the front face, therefore only this portion of the mesh from a head is needed. This gives an outermost contour which bounds all other three inner contours, as shown in the Face Contour in Figure 3.1.

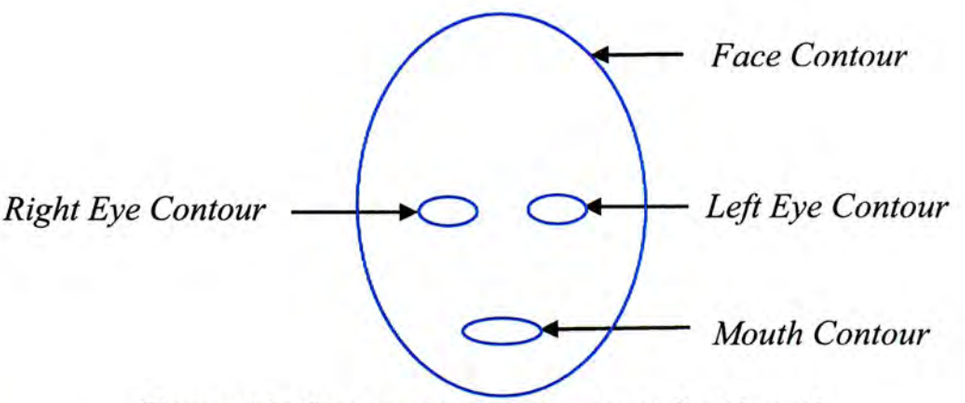


Figure 3.1. Four contours in the input facial mesh

3.1.3 Type of the polygons of the facial mesh

To simplify the implementation of the BEM engine, only triangular polygons are supported in the system. Therefore the input mesh is triangulated before it is passed to the system for processing.

3.2 Facial modeling and Feature recognition

The facial modeling process mainly involves a face offset step and a face-thickening step. The facial feature recognition process makes use of the topological property of the input mesh to locate the position of the eyes and the mouth. The system makes use of the locations of the facial features to determine the ten major facial muscles on the facial mesh.

3.3 User Control

User is allowed to specify the amount for the offset and the thickness during the facial modeling process. If the locations of the facial muscles obtained in the feature recognition process are not satisfactory, user is allowed to move and resize the bounding boxes of the facial features so as to change the locations of the facial muscles. When the BEM engine completes it process, ten muscle sliders as shown in Figure 3.3 are provided for controlling the movement of the facial muscles. Different facial expressions can be produced by using these muscle sliders.

3.4 Output of the system

The system outputs different facial expressions by evaluating a new solution using the result from the BEM engine when user moves the muscle sliders. The data of the

facial mesh can be exported as a data file that can be read by the other CAD software.

Figure 3.2 shows the overview of the Facial Expression System.

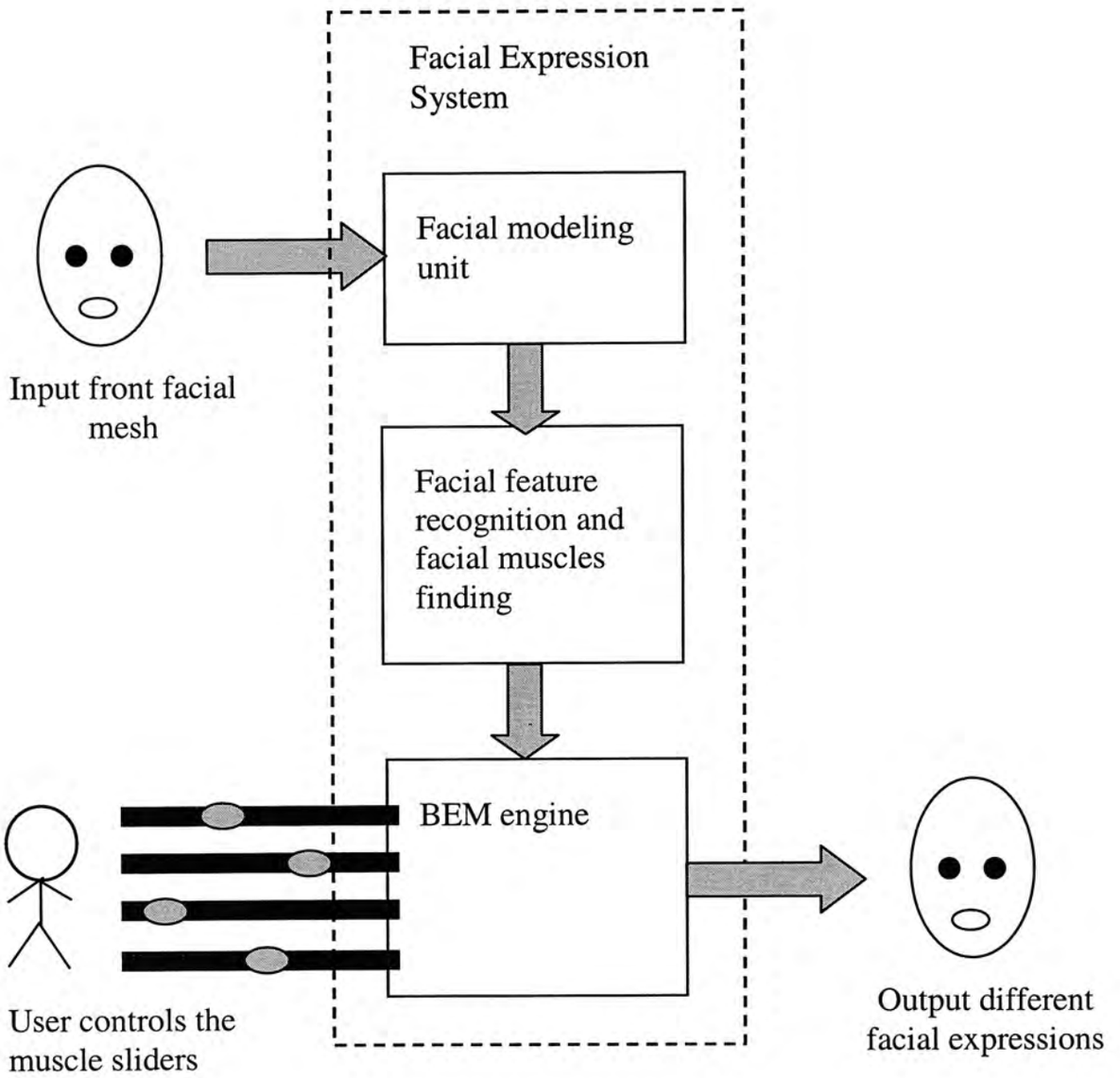


Figure 3.2. Overview of the Facial Expression System

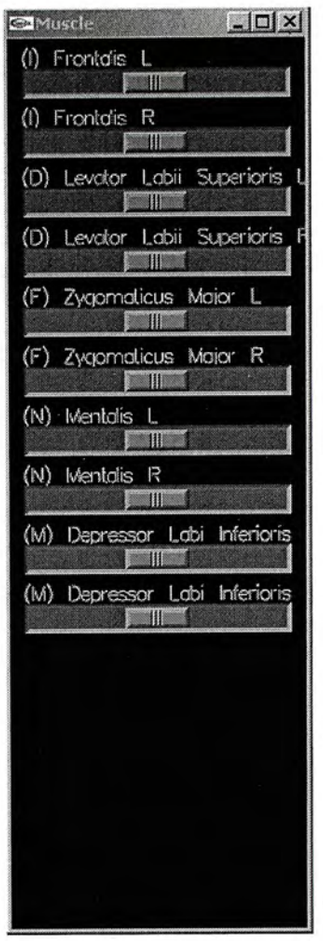


Figure 3.3 The ten muscle sliders

Chapter 4. Boundary Element Method (BEM)

To perform a linear elastic deformation for the facial mesh, the Facial Expression System makes use of the Boundary Element Method (BEM). To apply the BEM technique in the system, two types of boundary conditions must be specified. They are the displacement boundary condition, and the traction boundary condition. The displacement boundary conditions are the displacements controlled by the facial muscles and the skull. The traction boundary conditions are the zero tractions at locations where the polygons are free to move.

For a surface point $\boldsymbol{P}$ (field point) and a surface point $\boldsymbol{Q}$ (load point), the BEM governing equation for stress analysis is given by:

$$C(P)u(P) + \int_S T(P,Q)u(Q)dS_Q = \int_S U(P,Q)t(Q)dS_Q \qquad (4.1)$$

where $\boldsymbol{T(P,Q)}$ is the traction kernel and $\boldsymbol{U(P,Q)}$ is the displacement kernel. The numerical implementation of equation (4.1) is based on discretization of the boundary integrals. The boundary integrals of the BEM are discretized by using constant elements in the system.

The setup of the BEM engine using the constant element will be discussed in the following sections.

4.1 Numerical integration of the kernels

The constant element version of the BEM engine is implemented in the Facial Expression System. By using the constant element for the BEM implementation, each boundary element is represented by a polygon of the facial mesh. When the field point $\boldsymbol{P}$ and the load point $\boldsymbol{Q}$ are different, the integrals in equation (4.1) are not

singular and thus can be evaluated by using the standard Gaussian quadrature. When the field point ***P*** and the load point ***Q*** coincide, the integrals in equation (4.1) become singular and special techniques will be used to find them.

4.1.1 P and Q are different

In this case both the traction kernel and the displacement kernel are not singular. The discretized version of equation (4.1) using constant elements is:

$$C(P)u(P) + u(Q)\sum_{i=1}^{n} w_i T(P, Q(\xi_1^i, \xi_2^i, \xi_3^i)) = t(Q)\sum_{i=1}^{n} w_i U(P, Q(\xi_1^i, \xi_2^i, \xi_3^i)) \qquad (4.2)$$

where n is the total numbers of the integration points; ξ^i_1, ξ^i_2, ξ^i_3 are the coordinates of the i-th integration point and w_i is the weight for the Gaussian quadrature using the triangular coordinates.

Since the term ***C(P)u(P)*** only contributes to the diagonal 3x3 sub-matrix $\boldsymbol{h_{ii}}$ of the matrix ***H*** in equation (4.3), by using the rigid body transformation technique when finding the singular traction kernel, it needs not be calculated explicitly.

As using the constant element, each polygon will be iterated to setup the BEM engine. Taking the centroid of each polygon as the load point ***P***, and integrating the traction kernels and displacement kernels of equation (4.2), the integral equations reduce to a matrix form:

$$Hu = Gt \qquad (4.3)$$

where ***H*** is the square matrix of the traction kernel and ***G*** is the square matrix of the displacement kernel; ***u*** contains both the known and unknown displacement boundary conditions and ***t*** contains both the known and unknown traction boundary conditions. Both the ***u*** and ***t*** are column vectors. In the system all the known traction

boundary conditions are set to zero, which means those boundary elements are free to move.

The dimension of the matrix $\boldsymbol{H}$, $\boldsymbol{G}$ in equation (4.3), and $\boldsymbol{K}$ in equation (4.10) is $(3m)^2$, where m is the number of polygons of the facial mesh. In this stage, all the off-diagonal 3x3 sub-matrix $\boldsymbol{h_{ij}}$ and $\boldsymbol{g_{ij}}$ of the matrix $\boldsymbol{H}$ and $\boldsymbol{G}$ are found.

4.1.2 P and Q are identical

The evaluation of singular integrals is an essential part for the implementation of the BEM. The traction kernel and the displacement kernel become singular when the field point $\boldsymbol{P}$ and the load point $\boldsymbol{Q}$ coincide. By using a rigid body transformation, the singularity in the traction kernel can be solved. The singular displacement kernel can be found by transforming the surface integral into a line integral.

4.1.2.1 Evaluation of the Singular Traction Kernel

When the object undergoes a rigid body transformation, all elements are free to move and therefore there is no traction at all the surface nodes. Equation (4.3) becomes:

$$Hu = 0 \tag{4.4}$$

Under rigid body transformation, we can calculate the singular traction kernels as follows:

$$h_{ii} = -\sum_{j \neq i} h_{ij} \tag{4.5}$$

where $\boldsymbol{h_{ii}}$ is a 3x3 sub-matrix in the traction kernel matrix $\boldsymbol{H}$ in equation (4.3),.

After this step, all the diagonal 3x3 sub-matrix $\boldsymbol{h_{ii}}$ of the matrix $\boldsymbol{H}$ in equation (4.3) are found.

4.1.2.2 Evaluation of the Singular Displacement Kernel

By transforming the singular surface integral into a line integral along the three sides of the triangle, the integral can be solved analytically. Figure 4.1 shows the notations for the line integrals of equation (4.6). When integrating along the side v_1v_2, the value D is the length of cp_{12}; e_1 and e_2 are the direction cosines of the vector $v_2 - v_1$ and cp_{12}; θ_s and θ_r are the angles a_{112} and a_{212}, similar to the other two sides. δ_{ij} is the Kronecker delta function, μ is the shear modulus, and v is the Poisson's ratio.

After this step, all the diagonal 3x3 sub-matrix $\boldsymbol{g_{ii}}$ of the matrix $\boldsymbol{G}$ in equation (4.3) are found.

$$\int_{S^e} U_{ji} dA = DA \left[\delta_{ji}(3-4\nu)\ln(\tan\theta + \sec\theta) + a_{ji}\{\ln(\tan\theta + \sec\theta) - \sin\theta\} + b_{ji}\cos\theta + c_{ji}\sin\theta \right]_{-\theta_r}^{\theta_s} \tag{4.6}$$

where

$$A = \frac{1}{16\pi\mu(1-\nu)}$$

$$a_{ij} = e_{1i}e_{1j}$$

$$b_{ij} = -(e_{2i}e_{1j} + e_{1i}e_{2j})$$

$$c_{ij} = e_{2i}e_{2j}$$

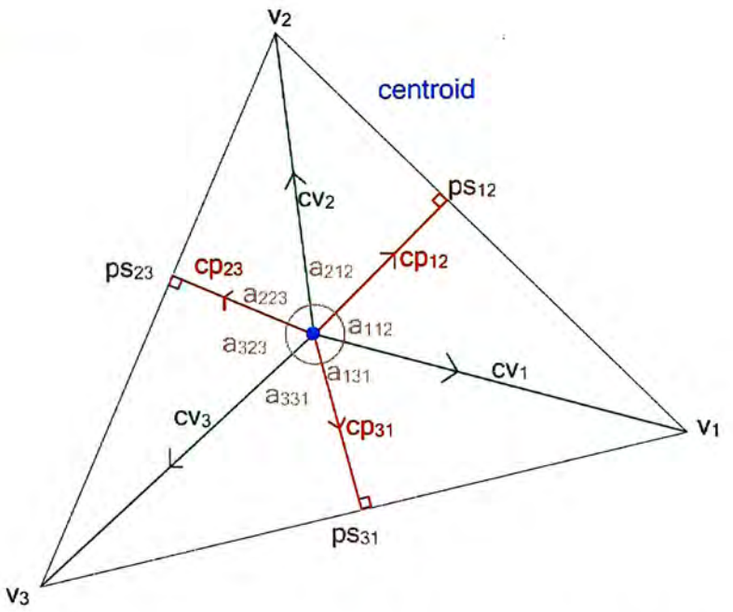


Figure 4.1. Notations of the line integrals for the singular displacement kernel

4.2 Assemble the stiffness matrix

After finding all the elements of the matrix $\boldsymbol{H}$ and $\boldsymbol{G}$ and applying all the prescribed boundary conditions in equation (4.3), we need to rearrange all the known boundary conditions (including the prescribed displacement boundary conditions and the zero traction boundary conditions) to the right hand side and assemble the stiffness matrix to the left hand side in order to find the solution.

Assume we have the prescribed displacement boundary conditions $\boldsymbol{u_2}^*$ and $\boldsymbol{u_4}^*$; the prescribed traction boundary conditions $\boldsymbol{t_1}^*$ and $\boldsymbol{t_3}^*$; and the unknown displacement boundary conditions $\boldsymbol{u_1}$ and $\boldsymbol{u_3}$; the unknown traction boundary conditions $\boldsymbol{t_2}$ and $\boldsymbol{t_2}$ as shown in equation (4.7).

To rearrange all the known boundary conditions to the right hand side, the columns of their corresponding coefficients in the matrix $\boldsymbol{H}$ will be swapped with the columns of their corresponding coefficients in the matrix $\boldsymbol{G}$ with a negative sign. Equation (4.7) shows the system of equations after the known displacement boundary condition $\boldsymbol{u_2}^*$ rearranged to the right hand side. Equation (4.8) shows the

system of equations after both the known displacement boundary condition u_2^* and u_4^* rearranged to the right hand side.

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & \cdots \\ h_{21} & h_{22} & h_{23} & h_{24} & \cdots \\ h_{31} & h_{32} & h_{33} & h_{34} & \cdots \\ h_{41} & h_{42} & h_{43} & h_{44} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} u_1 \\ u_2^* \\ u_3 \\ u_4^* \\ \cdots \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} & \cdots \\ g_{21} & g_{22} & g_{23} & g_{24} & \cdots \\ g_{31} & g_{32} & g_{33} & g_{34} & \cdots \\ g_{41} & g_{42} & g_{43} & g_{44} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} t_1^* \\ t_2 \\ t_3^* \\ t_4 \\ \cdots \end{bmatrix} \tag{4.7}$$

$$\begin{bmatrix} h_{11} & -g_{12} & h_{13} & h_{14} & \cdots \\ h_{21} & -g_{22} & h_{23} & h_{24} & \cdots \\ h_{31} & -g_{32} & h_{33} & h_{34} & \cdots \\ h_{41} & -g_{42} & h_{43} & h_{44} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} u_1 \\ t_2 \\ u_3 \\ u_4^* \\ \cdots \end{bmatrix} = \begin{bmatrix} g_{11} & -h_{12} & g_{13} & g_{14} & \cdots \\ g_{21} & -h_{22} & g_{23} & g_{24} & \cdots \\ g_{31} & -h_{32} & g_{33} & g_{34} & \cdots \\ g_{41} & -h_{42} & g_{43} & g_{44} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} t_1^* \\ u_2^* \\ t_3^* \\ t_4 \\ \cdots \end{bmatrix} \tag{4.8}$$

$$\begin{bmatrix} h_{11} & -g_{12} & h_{13} & -g_{14} & \cdots \\ h_{21} & -g_{22} & h_{23} & -g_{24} & \cdots \\ h_{31} & -g_{32} & h_{33} & -g_{34} & \cdots \\ h_{41} & -g_{42} & h_{43} & -g_{44} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} u_1 \\ t_2 \\ u_3 \\ t_4 \\ \cdots \end{bmatrix} = \begin{bmatrix} g_{11} & -h_{12} & g_{13} & -h_{14} & \cdots \\ g_{21} & -h_{22} & g_{23} & -h_{24} & \cdots \\ g_{31} & -h_{32} & g_{33} & -h_{34} & \cdots \\ g_{41} & -h_{42} & g_{43} & -h_{44} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} t_1^* \\ u_2^* \\ t_3^* \\ u_4^* \\ \cdots \end{bmatrix} \tag{4.9}$$

After grouping all the known boundary conditions to the right hand side, the BEM system becomes:

$$Kx = y \tag{4.10}$$

where $\boldsymbol{K}$ is the square stiffness matrix, $\boldsymbol{x}$ contains all the unknown boundary conditions and $\boldsymbol{y}$ contains all the known boundary conditions. Having equation (4.10), we solve the BEM system (i.e. find the inverse of the matrix $\boldsymbol{K}$) by the Gauss Elimination. After solving the BEM system, all the displacement boundary conditions are known, therefore all the polygons of the facial mesh can be updated accordingly.

Chapter 5. Facial Modeling

The input to the system is a triangular polygonal mesh consisting of an opened mouth and two opened eyes. This input mesh is considered as the front face of the facial mesh. Since the front face mesh is an open surface, and the BEM engine in the system requires a closed surface, special step is needed to transform the input front face mesh into a closed surface mesh. To obtain a closed surface mesh from a single sided front face mesh, an offset and a polygon insertion processes are used. The details of these two steps will be discussed below.

5.1 Offset of facial mesh

The first step of the facial modeling process is to offset the front face mesh to obtain a back face mesh. The main purpose of the step is to generate a closed surface that can be used in the BEM engine. The back face mesh will have the same number of polygons and same topology as the front face mesh. Both of them share the same four contours of the given facial mesh. The vertices which belong to those four contours are fixed, while other vertices will be moved during the offset process. To offset the front face mesh, user is allowed to enter the amount for the offset. Figure 5.1 shows the front face mesh and the back face mesh. Figure 5.2 combines the two face mesh together.

Before the offset process, a copy of the front face mesh will be cloned to be the back face mesh. The offset process will be operated on the back face mesh. Let V_o be the set that contains the back face mesh vertices which does not belong to the four contours. The offset process moves each vertex $v_i \in V_o$ in its normal direction.

The vertex normal is calculated by averaging the normals of its neighboring polygons.

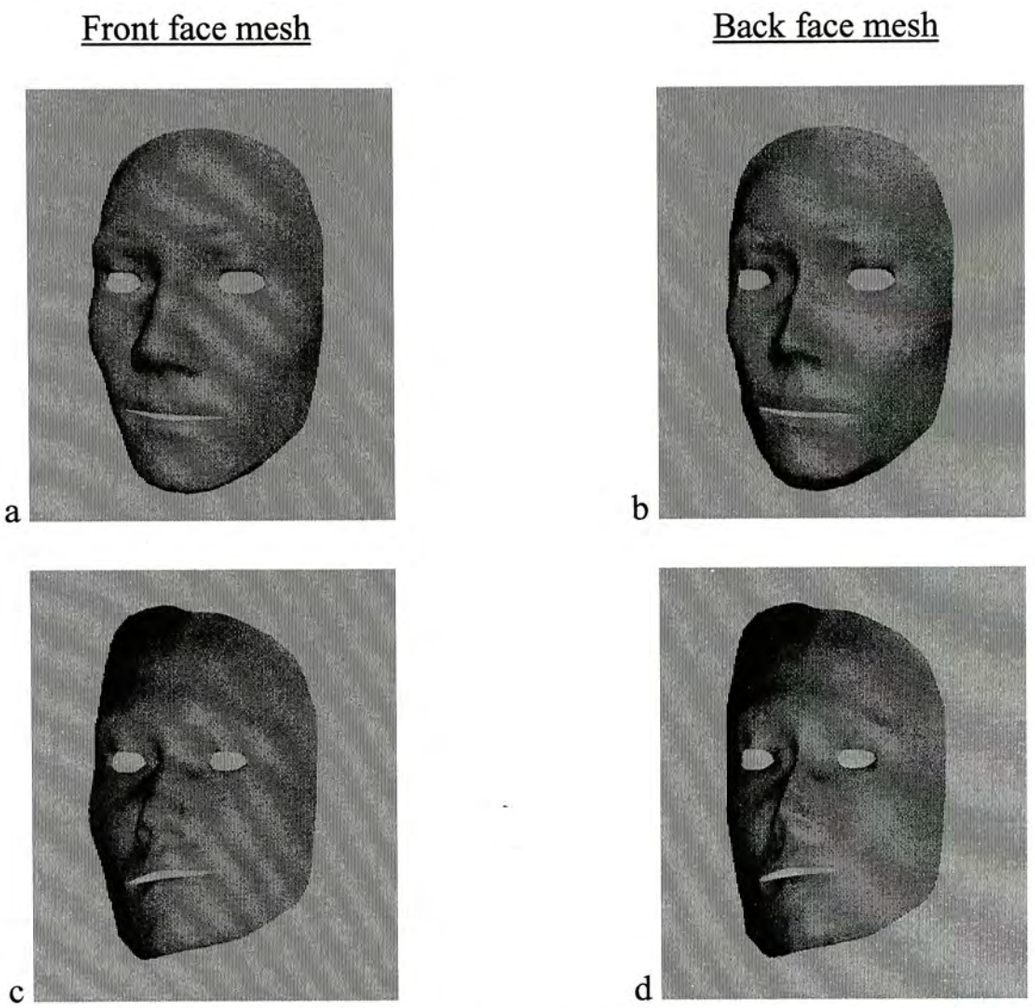


Figure 5.1. a, c) Front and rear views of the input front face mesh.

b, d) Front and rear views of the back face mesh

a  b 

Figure 5.2. a) The input front face mesh is shown in white wireframe, and the back face mesh is shown in green wireframe. b) The shaded view of the front and back face mesh

5.2 Thickening of Face Contour

The facial mesh offset are of a specified thickness at the inner region whereas the four contours are of no thickness. The last step of the facial modeling process is to thicken the *Face Contour* by inserting some new polygons between the front and back face mesh. As the *Face Contour* is an imaginary edge rather than real edge on a human face, a thickness defined will provide constraint on the *Face Contour* which keeps the contour immovable. Each node or each boundary element for a constant element BEM implementation is represented by a polygon. Therefore, the newly inserted polygons on the *Face Contour* will be a part of the fixed displacement boundary conditions. The newly created polygons on the *Face Contour* are shown in Figure 5.3 and Figure 5.4.

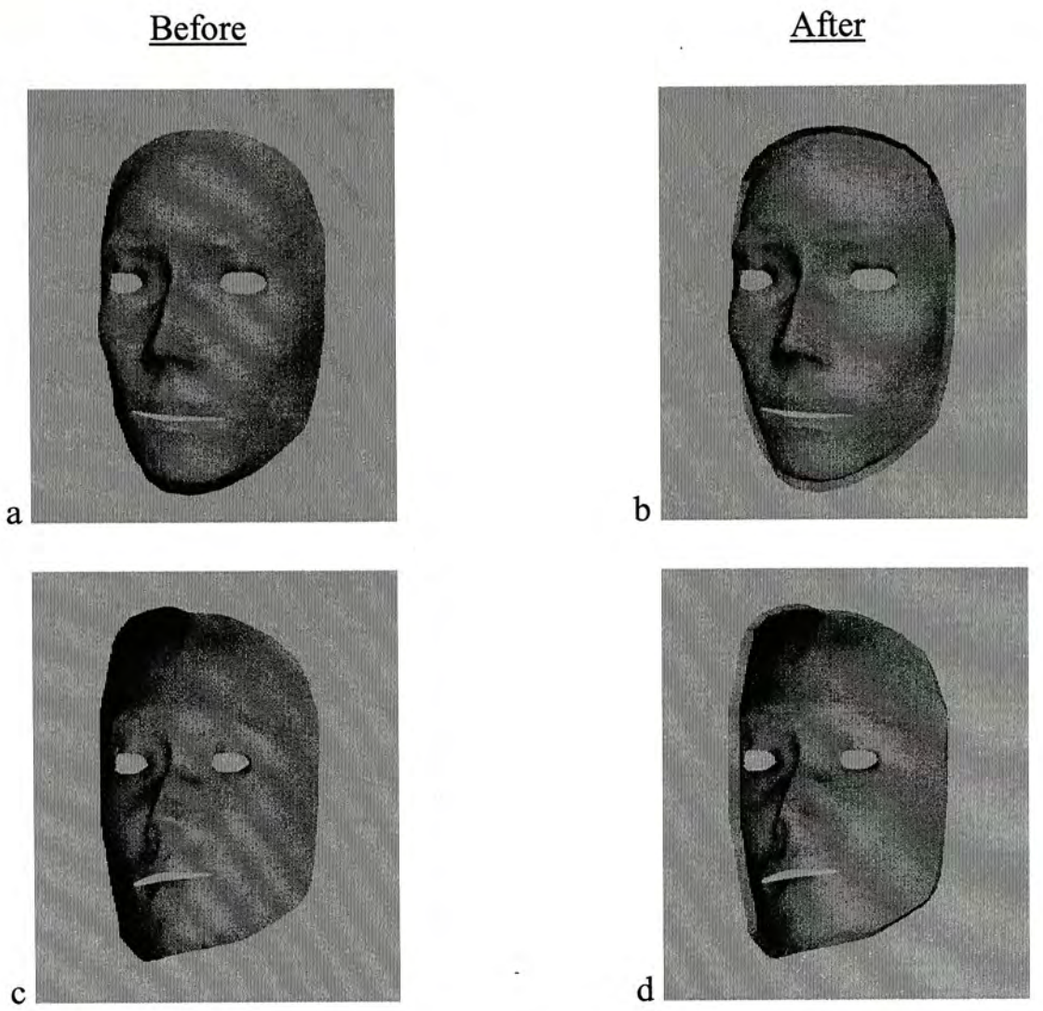


Figure 5.3. a, c) back face mesh after the offset. b, d) back face mesh after the thickening the *Face Contour*

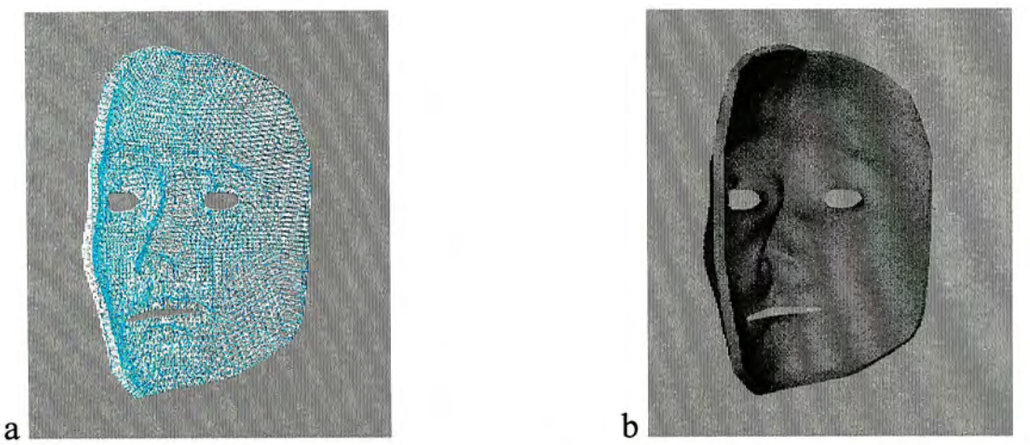


Figure 5.4. a) The input front face mesh is shown in white wireframe, and the back face mesh is shown in green wireframe after the thickening step. b) The shaded view of the front and back face mesh after the thickening step.

Chapter 6. Facial Feature Recognition

In order to locate the facial muscles on the facial mesh, the positions of the eyes, the nose and the mouth must be known. In the system, the topology of the facial mesh is used to recognize the location of these facial features.

Since the eyes and the mouth are holes in the input facial mesh, there is enough information to locate these facial features. There are several steps to extract the facial features from the facial mesh, they are 1) extracting all contour edges, 2) separating different holes from the contour edges, 3) finding the bounding boxes of different holes, and 4) determining those facial features based on the bounding boxes.

After the facial feature recognition process, *Feature Boxes* (which are actually the bounding boxes of the facial features) are located. Those *Feature Boxes* will be used later to find the facial muscles (for specifying boundary conditions for the BEM engine). There are six *Feature Boxes* in the system; one for the face (*FBFace*), two for the eyes (*FBEyeL* and *FBEyeR*), one for the mouth (*FBMouth*), one for the nose (*FBNose*), and the other for the skull (*FBSkull*). The first four *Feature Boxes* are generated in the facial feature recognition process, *FBNose* and *FBSkull* are calculated based on the *Feature Boxes FBEyeL, FBEyeR and FBFace*. Section 6.4 discusses the locations of those *Feature Boxes*.

Since the sizes of those *Feature Boxes* affect the facial muscles locations on the facial mesh, user is allowed to grow or shrink those *Feature Boxes* after the facial feature recognition process.

The details of each step will be discussed in the following sections.

6.1 Extract all contour edges from the facial mesh

As shown in Figure 6.1, the facial mesh consists of four independent contours, they are the *Face contour*, *Left Eye Contour*, *Right Eye Contour* and the *Mouth Contour*. In this step, the contour edges ce_{ij} (edges used by only one polygon) are extracted from the facial mesh into a set of contour edges $E_{contour}$. It is obvious that all elements of $E_{contour}$ are edges of the four independent contours in the facial mesh.

Let E be a set that contains all the edges of the facial mesh. To extract the contour edges, each edge $e_{ij} \in E$ is examined, if the number of adjacent polygons of an edge e_{ij} is one, it is a contour edge and it will be inserted into $E_{contour}$.

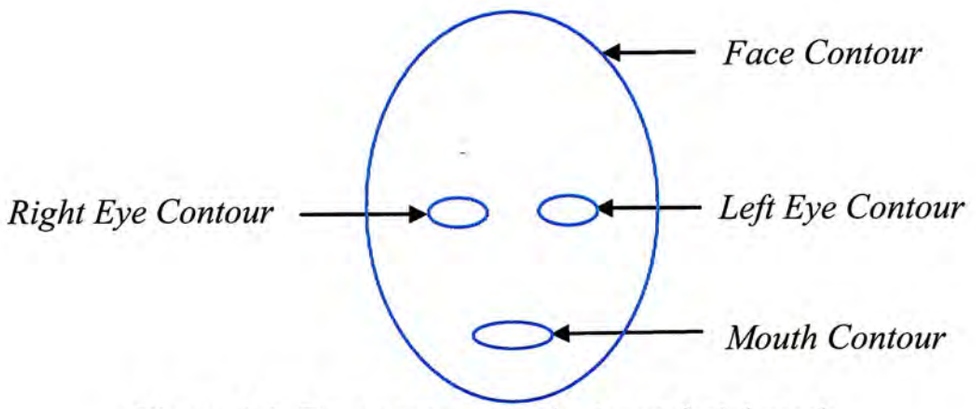


Figure 6.1. Four contours in the input facial mesh

6.2 Separate different holes from the contour edges

The four holes in the facial mesh can be extracted after the set $E_{contour}$ is found. Since a hole in the facial mesh is constructed by a series of contour edges $ce_{ij} \in E_{contour}$ which connected in a loop form, exploring this connectivity property leads to the solution. Four holes in the facial mesh are the four independent connected loops in $E_{contour}$. However there is no information to distinguish whether a contour is a face, eye, or mouth contour, therefore only four holes are found after this step. They are called H_i where $i = 0, 1, 2, 3$.

To extract the first hole H_0 from the set $E_{contour}$, I start extracting the first contour edge $ce_{ij} \in E_{contour}$ and it is inserted into the set H_0. While the first and the last elements are not connected in H_0, each contour edge $ce_{ij} \in E_{contour}$ is examined and if it is connected with the last element in the set H_0, it will be inserted into H_0. When all ce_{ij} are examined, the set H_0 will contain a series of contour edges constitute one of the four contours of the facial mesh.

To extract the other hole H_i where $i \neq 0$ from the set $E_{contour}$, I choose a contour edge $ce_{ij} \in E_{contour}$ which is not contained in any other set H_j where $j \neq i$ and insert it into the set H_i. While the first and the last elements are not connected in H_i, each contour edge $ce_{ij} \in E_{contour}$ will be examined and if it is connected with the last element in the set H_i, it is inserted into H_i. When all ce_{ij} are examined, the set H_i will contain a series of contour edges forming one of the remaining three unknown contours of the facial mesh. This procedure is repeated until all the three hole sets H_1, H_2, and H_3 are extracted.

In this stage, all four holes are separated from the contour edges set $E_{contour}$.

6.3 Locating the bounding boxes of different holes

After extracting all the four holes, locating the bounding boxes of these holes is straightforward. However this step is essential in the sense that the bounding boxes of these holes will be used to determine the facial features in the next step.

To calculate the bounding box of a hole, each edge in the hole set H_i where $i = 0, 1, 2, 3$ is iterated. By examining the coordinates of the vertices of the edge, the four bounding boxes $BBox_i$ where $i = 0, 1, 2, 3$ can be found.

6.4 Determine the facial features

By comparing the human face and the positions of the bounding boxes of the four holes, the facial features can easily be identified. Five properties of the human face are used to determine the facial features in the system, as shown in Figure 6.2, they are 1) the left eye must be located on the left side of the face, 2) the right eye must be located on the right side of the face, 3) the mouth must be located on the lower portion of the face, 4) the ratio a : b is approximately 1 : 1, and 5) the width w of the nose is roughly the same as the separation between the two eyes. By examining all the eight corners of a bounding box and using the properties above, the four facial features left eye, right eye, mouth, and the face can be distinguished among the four bounding boxes.

Figure 6.3 shows all the six Feature Boxes extracted from an input front facial mesh.

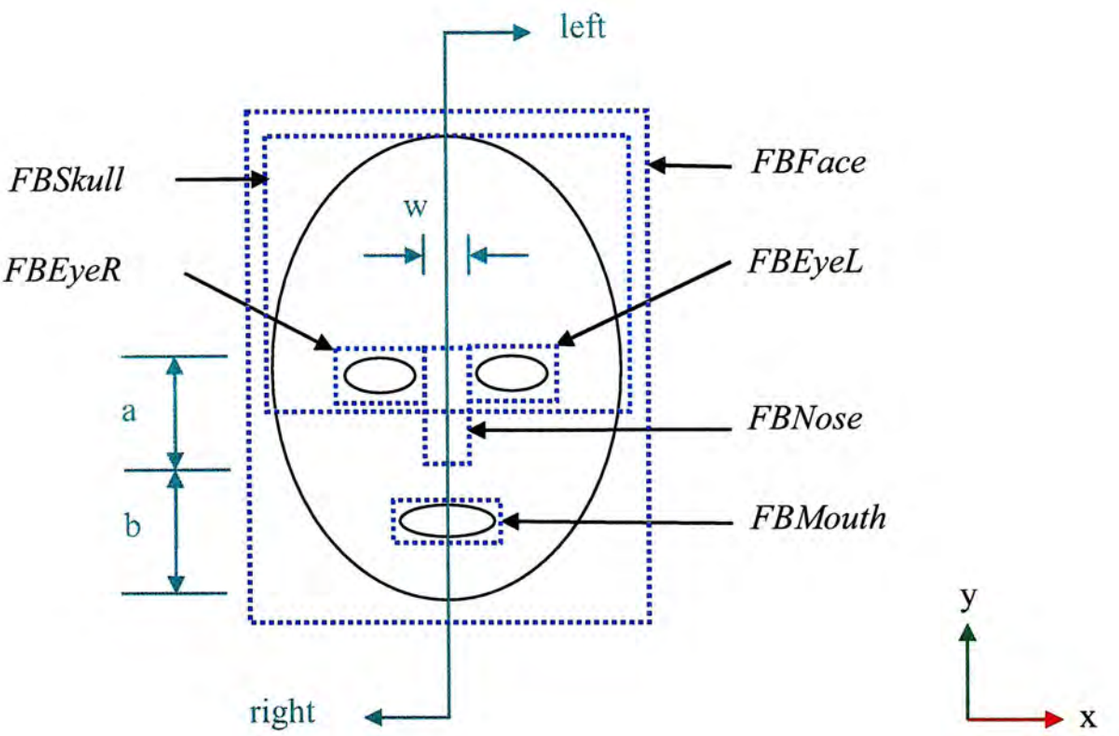


Figure 6.2. The six *Feature Boxes* of a facial mesh

6.4.1 Eye positions

If all the eight corners of a bounding box are located on the left half of the face, it is the bounding box of the left eye. Similarly for the right eye, if all the eight corners of a bounding box are located on the right half space of the face, it is the bounding box of the right eye.

6.4.2 Mouth position and Face

There are only two bounding boxes left to be identified, and they must be the bounding box of the mouth and the face. Since the mouth is assumed to be located in the lower portion of the face, the bounding box of the mouth can be identified by comparing the two centroids of the bounding boxes. The bounding box of the mouth is the one having lower centroid coordinates.

6.4.3 Nose position

The bounding box of the nose is calculated by the minimum separation w between the two eyes' bounding boxes, and the ratio between a and b in Figure 6.2.

6.4.4 Skull position

The bounding box of the skull is calculated based on the bounding boxes of the face and the eyes. It covers the upper region of the bounding box of the face. It represents the forehead of a human face.

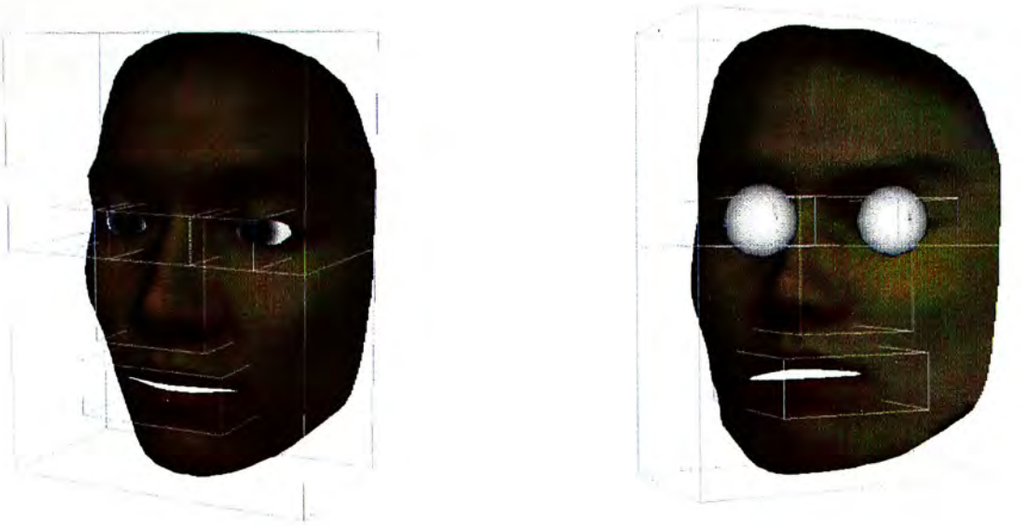


Figure 6.3. The *Feature Boxes* found from an input facial mesh

Test results show that the performance of the feature recognition is good. It always gives the correct result in a human face. Since the adjacency information of the edges is stored in the system, the computation complexity in finding the contours edges is O(e) where e is the number of edges of the facial mesh. The feature recognition processes take $O(c^2)$ where c is the number of contour edges of the facial mesh.

Our feature recognition algorithm assumes that the input facial mesh has been axis aligned. If we allow arbitrary facial mesh orientation, the algorithm has to be revised to align the facial mesh with the system coordinates. This is one of the extensions of this project and it will be further discussed in the conclusions.

Chapter 7. Boundary Conditions in the system

In order to apply the Boundary Element Method for the deformation, some traction boundary conditions or some displacement boundary conditions must be specified. In the Facial Expression system, only the displacement boundary conditions are needed. Fortunately all the displacement boundary conditions are automatically generated by the system. If the result is not satisfied, tools are provided to tweak the output displacement boundary conditions.

There are two types of displacement boundary conditions in the system, one is called the zero displacement boundary conditions and the other is called the adjustable displacement boundary conditions. The zero displacement boundary conditions are applied to the nodes of the facial mesh that are attached to the skull. The adjustable displacement boundary conditions are the nodal displacement induced by the motion of facial muscles lying beneath the facial mesh. Varying the adjustable displacement boundary conditions causes the BEM engine to output a new set of solution. Therefore different facial expressions can be obtained by varying the adjustable displacement boundary conditions, which is the same as changing the shape of the facial muscles in the system.

There are two approaches for applying the displacement boundary conditions. The main difference of these two methods is on which part of the facial mesh the fixed displacement boundary conditions is applied. The first approach defines the entire back face mesh as the skull mesh, and the muscles are only defined on the front face mesh. The second approach takes only the regions of forehead, the regions around eyes and the region of nose from the backside as the skull mesh, and the facial muscles are defined on both the front and the back face mesh.

Test results show that the second approach works better than the first approach in the sense that the second approach allows greater movement on the cheek area where there is no bone behind it which is exactly what the second approach assumes. Besides, the mouth can be opened by contracting the mentalis and the levator labii superioris which are located around the lips. Since the result works quite similar to a human face, the second approach is adopted.

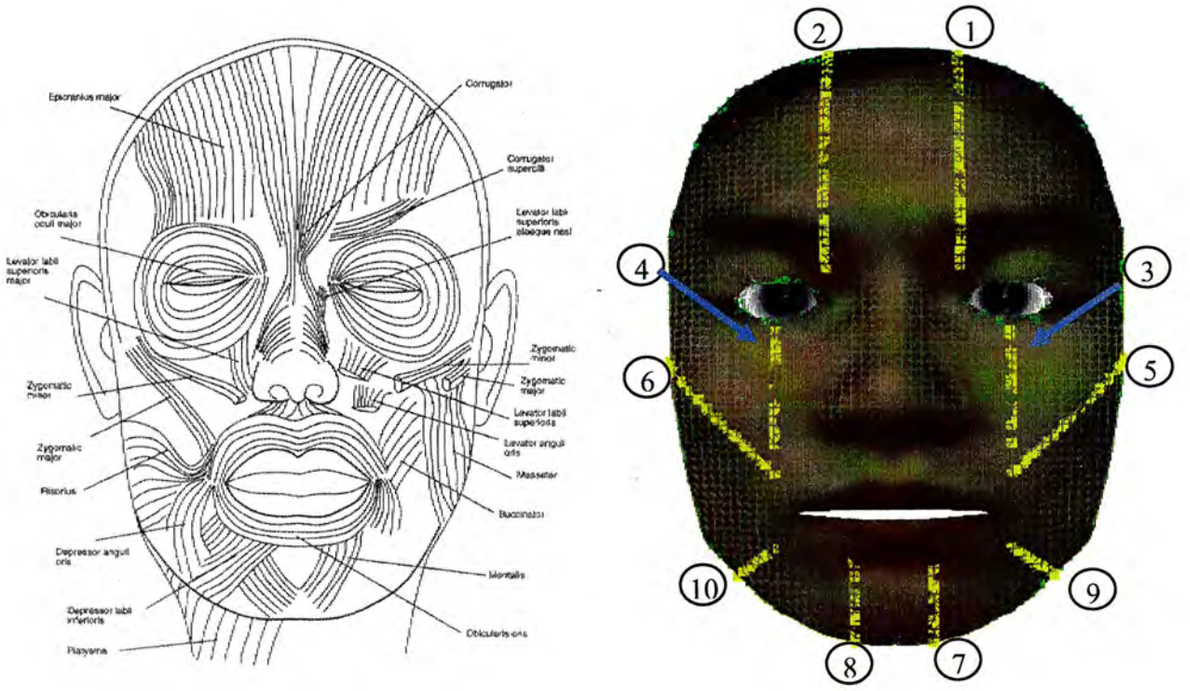
In the following sections the locations of the adjustable displacement boundary conditions and the locations of the fixed displacement boundary conditions will be shown, the details of the facial muscle recognition and the tool to tweak the facial mesh generation will also be discussed.

7.1 Facial Muscles

There are ten muscles to control the facial expression in the system, as shown in Figure 7.1, they are

Name of muscles	Functions [26]
1. Frontalis L 2. Frontalis R	raise the eyebrows, as in surprised
3. Levator Labii Superioris L 4. Levator Labii Superioris R	raise the upper lip, deepening the masolabial furrows like the levator labii superioris alaeque nasi
5. Zygomaticus Major L 6. Zygomaticus Major R	elevate the modiolus and buccal angle, as in laughing
7. Mentalis L 8. Mentalis R	elevate the skin of the chin aiding its protrusion/eversion, as in drinking
9. Depressor Labi Inferioris L 10. Depressor Labi Inferioris R	pull the lower lip down and laterally in mastication

The ten muscles selected are the major facial muscles that are sufficient in producing the primordial expressions. All of them are defined as the adjustable displacement boundary conditions. The notations 'L' and 'R' at the end of the name of the facial muscles represent the left side and the right side of the facial mesh. Among all the facial muscles, only the Frontalis L and the Frontalis R muscles consist of the front side polygons only, while all the other facial muscles consist of both the front and the back polygons.



a b

Figure 7.1. a) The facial muscles on a human face [26]

b) The polygons covered by ten facial muscles are shown with orange square

7.2 Skull Bone

There is only one skull bone in the system and it represents the fixed displacement boundary conditions. The skull bone covers the upper region of the head starting

from the nose, and the polygons next to the *Face Contour* of the facial mesh. Figure 7.2 shows the polygons covered by the skull bone.

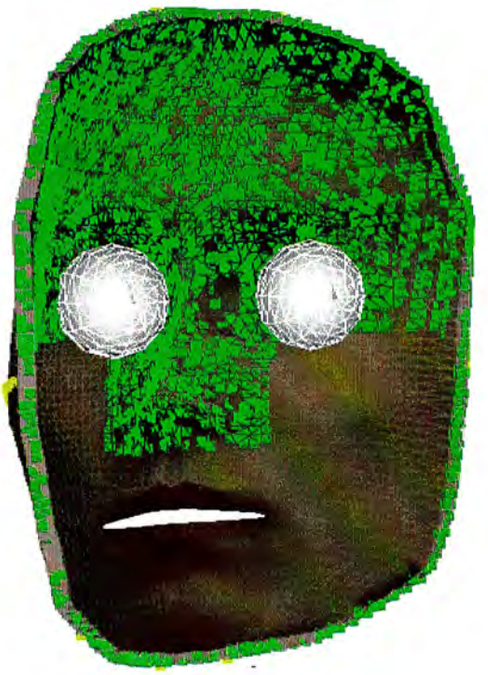


Figure 7.2. The polygons covered by the skull bone are shown with green square

7.3 Facial Muscle recognition

Before the facial muscles can be located on the facial mesh, the locations of some facial features must be known, and this has been discussed in chapter six. These facial features are bounded by the corresponding *Feature Boxes*.

There are two steps in locating the facial muscles in the system. In the first step, ten *muscle-definers* will be calculated based on the *Feature Boxes*. In the second step these ten *muscle-definers* are projected onto the facial mesh to locate the facial muscles.

7.3.1 Locating muscle-definers

The ten *muscle-definers* all lie on a plane which is parallel to the front side of the facial mesh. The numbering of the *muscle-definers* in the Figure 7.3 is the same as the numbering of the corresponding facial muscles in the previous section.

The coordinates of the ten *muscle-definers* are listed in Table 7.1 and their relative positions are shown in Figure 7.3.

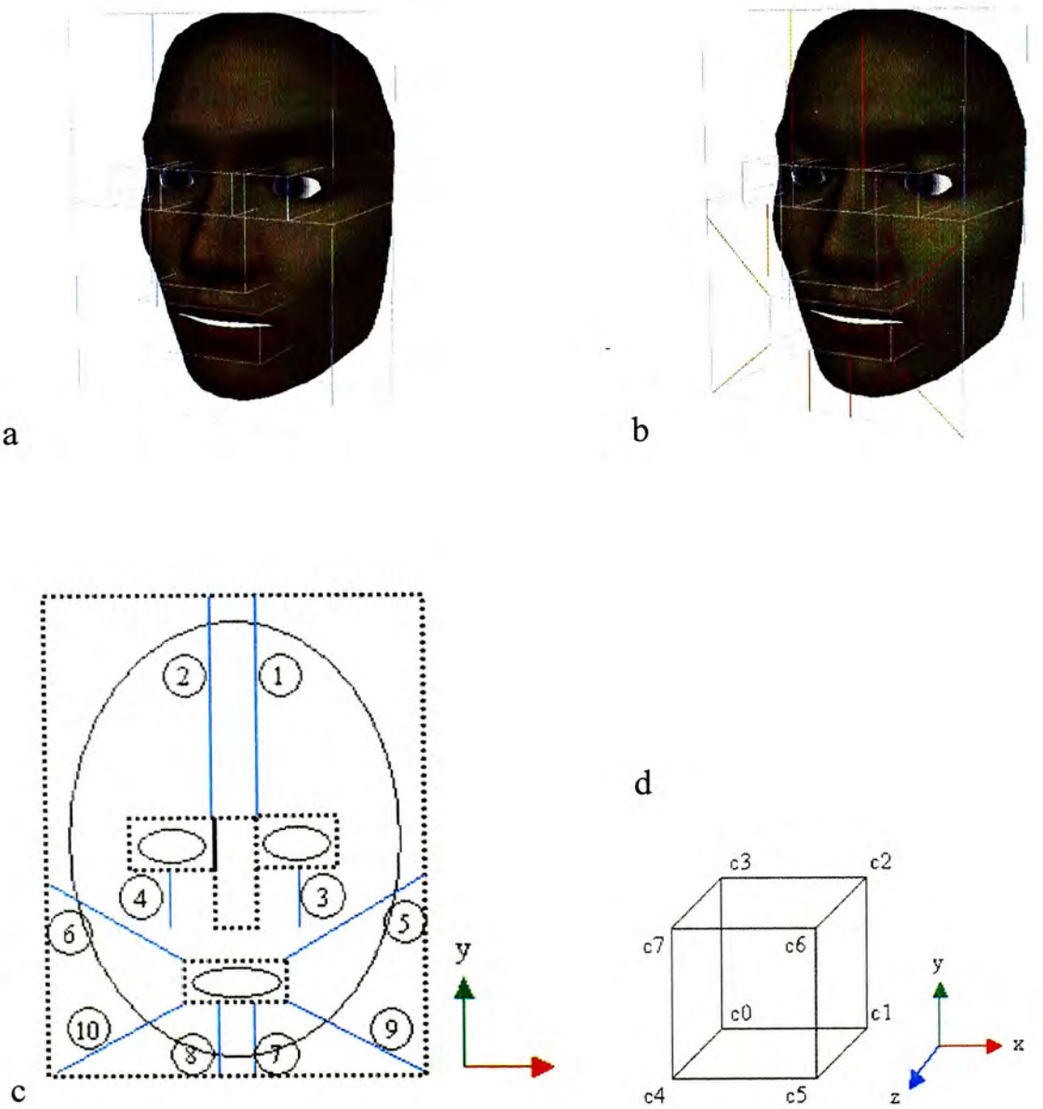


Figure 7.3. a) Locations of the *Feature Boxes*. b) Ten *muscle-definers* are shown in red. c) Locations of the ten *muscle-definers* in the system. d) The eight corner indexes of a bounding box

Muscle-definer	Start point	End point
1. Frontalis L	corner c6 of *Feature Box FBEyeL*	x-coordinate: same as start point y-coordinate: top of *Feature Box FBFace*
2. Frontalis R	corner c7 of *Feature Box FBEyeR*	x-coordinate: same as start point y-coordinate: top of *Feature Box FBFace*
3. Levator Labii Superioris L	x-coordinate: same as end point y-coordinate: bottom of *Feature Box FBNose*	mid-point between corners c4 and c5 of *Feature Box FBEyeL*
4. Levator Labii Superioris R	x-coordinate: same as end point y-coordinate: bottom of *Feature Box FBNose*	mid-point between corners c4 and c5 of *Feature Box FBEyeR*
5. Zygomaticus Major L	corner c7 of *Feature Box FBMouth*	x-coordinate: left of *Feature Box FBFace* y-coordinate: mid-point of *Feature Box FBNose*
6. Zygomaticus Major R	corner c6 of *Feature Box FBMouth*	x-coordinate: right of *Feature Box FBFace* y-coordinate: mid-point of *Feature Box FBNose*
7. Mentalis L	one-third of width of *Feature Box FBMouth* right from corner c5	x-coordinate: same as start point y-coordinate: bottom of *Feature Box FBFace*
8. Mentalis R	one-third of width of *Feature Box FBMouth* left from corner c4	x-coordinate: same as start point y-coordinate: bottom of *Feature Box FBFace*
9. Depressor Labi Inferioris L	corner c5 of *Feature Box FBMouth*	corner c5 of *Feature Box FBFace*
10. Depressor Labi Inferioris R	corner c4 of *Feature Box FBMouth*	corner c4 of *Feature Box FBFace*

Table 7.1. Locations of the ten *muscle-definers*

7.3.2 Locating muscles

The ten facial muscles are found by projecting the ten *muscle-definers* onto the facial mesh in the negative z direction and they are defined over the polygons of the facial mesh. The projection is done by subdividing the muscle-definer into segments, and then projecting the vertices of the segments onto the facial mesh to find the polygon of the corresponding facial muscle. The ordering of the polygons of a muscle is thus maintained. The steps in this process are listed below.

1. Subdivide the muscle-definer into segments.
2. For each vertex of the each segment, construct a ray originating from the vertex extending to the negative z direction (blue line in figure in Figure 7.4 a and b).
3. Intersect the ray with the facial mesh (intersected point is shown in red in Figure 7.4 a and b).
4. If the intersected polygon has not been processed, store it as a polygon of the corresponding muscle.
5. The process stops when all vertices of the segments have been processed and the muscle is found as is shown in light gray in Figure 7.4 b.

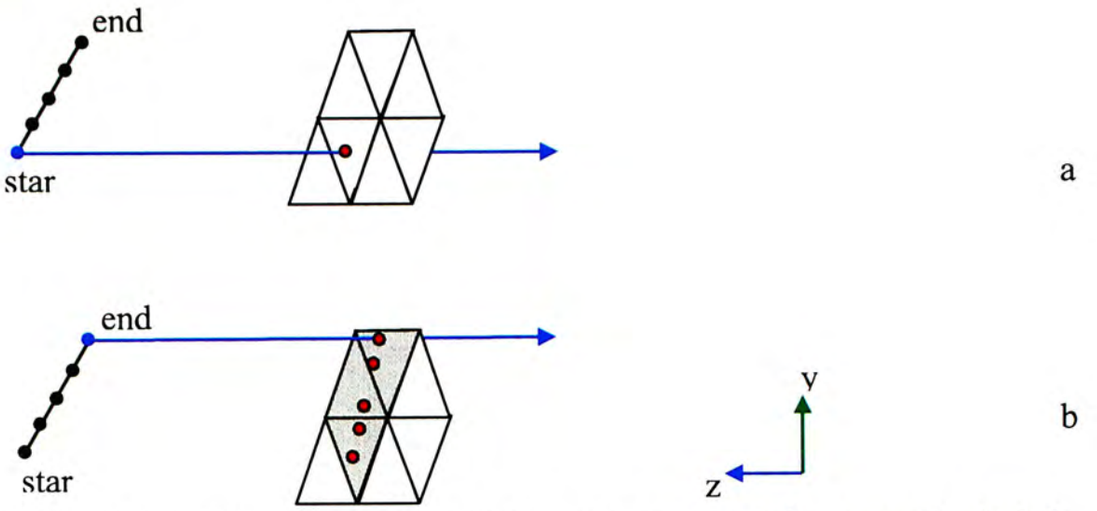


Figure 7.4. Facial muscle is found by projecting the *muscle-definer* onto the facial mesh

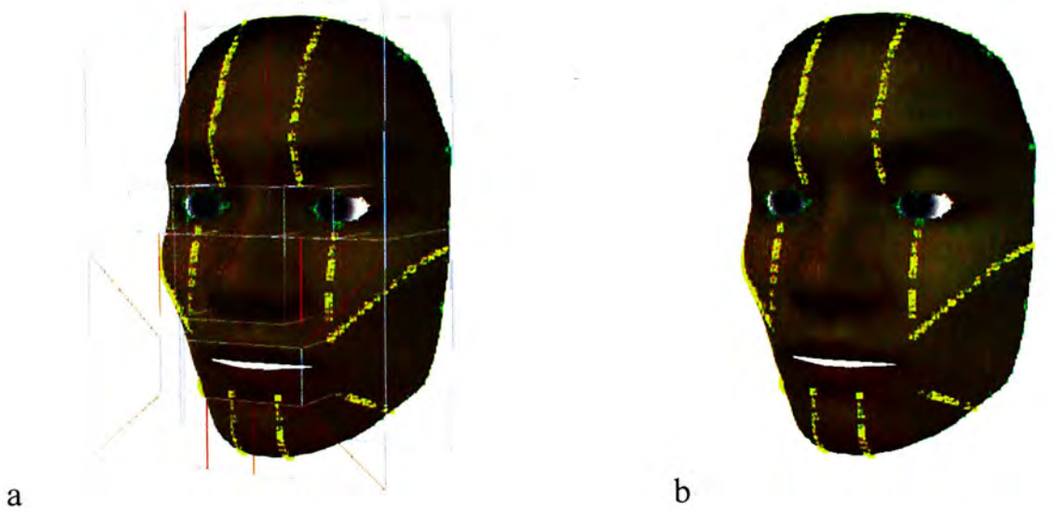
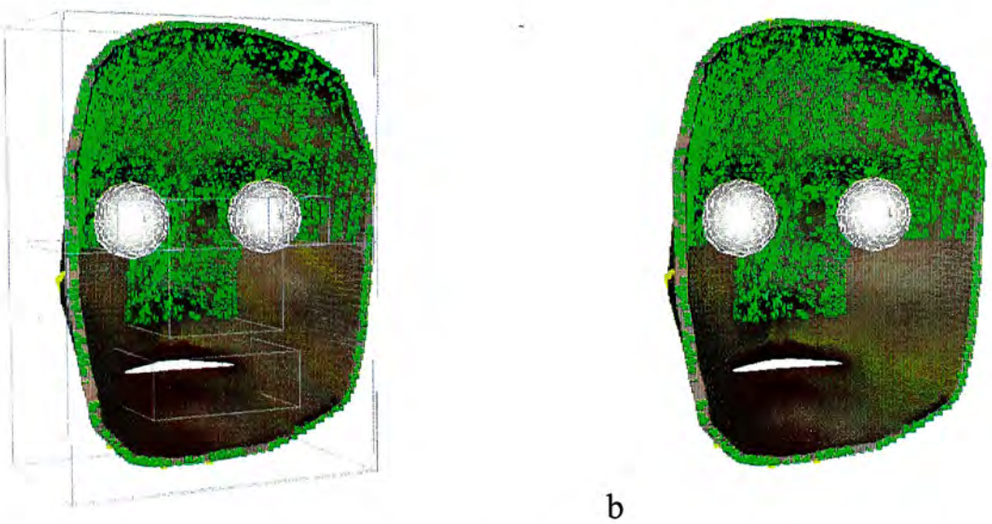


Figure 7.5. a) Relations between the *Feature Boxes*, the muscle-definers, and the Facial Muscle. b) Facial Muscles found by the system.

7.4 Skull Bone Recognition

The skull bone is found based on the *Feature Boxes FBSkull* and *FBNose* and the information stored during the facial modeling process. The polygons which are next to the Face Contour are all considered as a part of the skull bone (fixed displacement boundary conditions). The other part of the skull bone will be searched within the new polygons from the backside of the facial mesh.

To determine which polygons of the back face mesh are parts of the skull bone, all of their centroids are tested if they lie in the *Feature Boxes FBSkull* and *FBNose*. If the centroid of a polygon is inside either one of the two feature boxes, the polygon is considered as a part of the skull bone.



a b

Figure 7.6. a) Relations between the *Feature Boxes* and the Skull bone. b) Skull bone found by the system.

7.5 Refine the bounding regions of the facial features

As shown in Figure 7.3, the position and the size of the *Feature Boxes* determine the locations of the muscle-definers. To provide a better control over the facial muscle recognition, the size of the *Feature Boxes* can be changed by the user. This changes the locations of the muscle-definers, and the facial muscles will be changed as well.

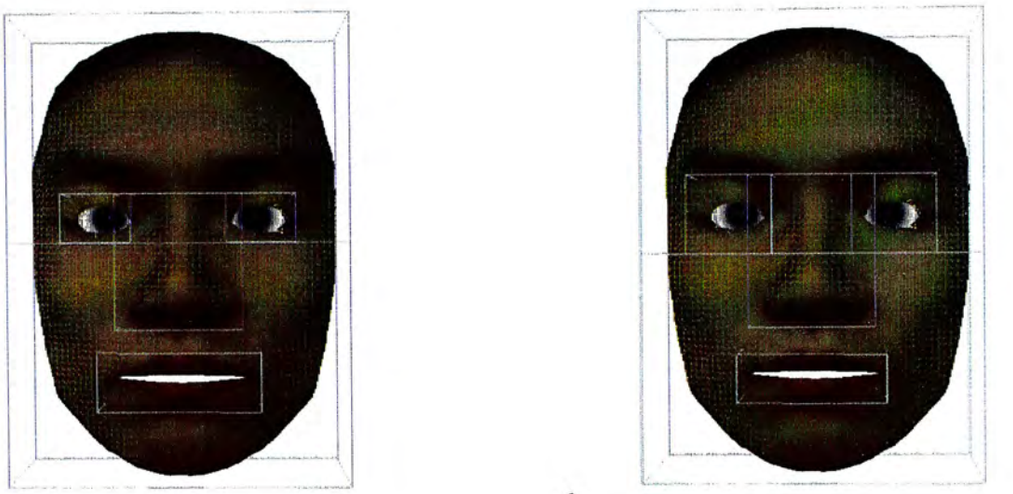
Figure 7.7 shows the differences before and after the resizing process. The *FBEyeL* and *FBEyeR* in Figure 7.7b are resized to be bigger than those in Figure 7.7a, Figure 7.7f shows that the facial muscles next to the eyes are shorter than those in Figure 7.7e.

7.6 Add/Remove facial muscles

The ten facial muscles recognized in the system are stored as a series of polygon indexes in a text file called muscle file. A facial muscle entity in the muscle file consists of the name, the size and the series of polygon indexes of a facial muscle as shown in Figure 7.8a. User can easily remove any of the ten predefined facial muscles easily by editing the muscle file. To add other facial muscle into the system, user just needs to add another facial muscle entity into the muscle file. The system has provided a polygon selection tool to output the polygon index as shown in Figure 7.8b. By using the selection tool and editing the muscle file manually, user can even change the recognized facial muscles. These processes allow flexible muscles assignments and enable expected deformation results.

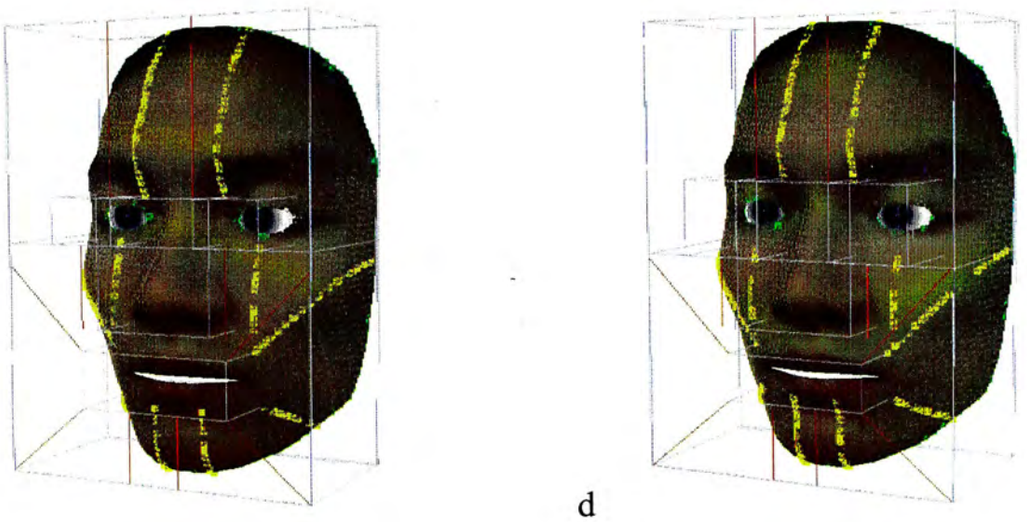
Before

After



a

b



c

d

e

f

Figure 7.7. a, c, e) the default facial muscles found by the system.

b, d, f) the facial muscles found by the system after resizing the *Feature Boxes*.

a

```
[Name]
(D) Levator Labii Superioris R
[Size]
8
[Data]
535
534
518
519
510
511
504
503
```

b

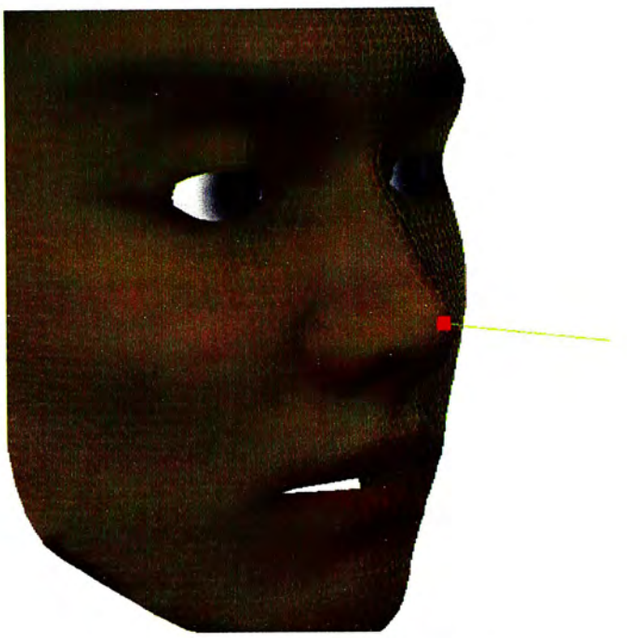


Figure 7.8 a) the data of a facial muscle stored in the muscle file. b) yellow ray is used to select the polygon of the facial mesh, red square is shown on top of the selected polygon.

Chapter 8. Muscles Movement

Human produce different facial expressions by changing the shape of the facial muscles. Similarly by deforming the polygons of the facial muscles, different facial expressions can be obtained. There are two types of deformation for the facial muscles in the system, they are the muscle contraction and the muscle relaxation. There are ten muscle sliders to control the deformations of the facial muscles interactively in the system. Details of the two types of muscle deformations and the muscle slider will be discussed in the following sections.

8.1 Muscle Contraction

The first type of muscle deformation in the system is the muscle contraction, it causes the polygons of a facial muscle to move towards the end point of its muscle while its end point is fixed. During the contraction, the polygons of the muscle will be deformed in the direction defined by the line connecting the centroids of the polygons. Figure 8.1 shows the facial muscle before and after the contraction.

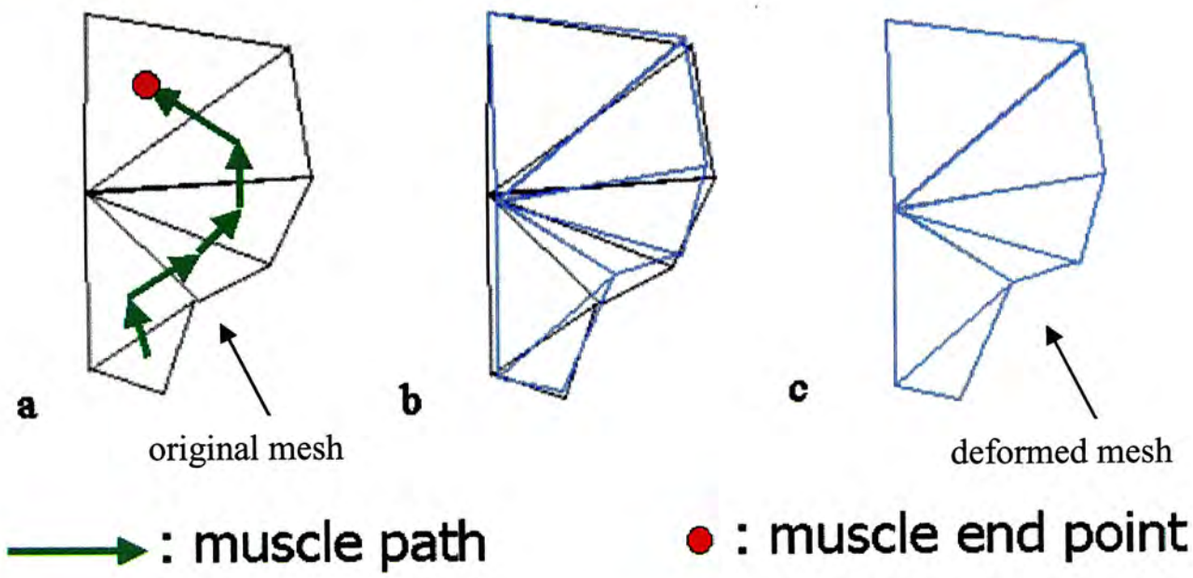


Figure 8.1. a) a facial muscle in normal state. b) overlapping of two facial muscles. c) a facial muscle after contraction.

8.2 Muscle Relaxation

The other type of muscle deformation in the system is the muscle relaxation, it causes the polygons of a facial muscle to move away from the end point of its muscle while its end point is fixed. During the relaxation, the polygons of the muscle will be stretched in the direction defined by the line connecting the centroids of the polygons. Figure 8.2 shows the facial muscle before and after the relaxation.

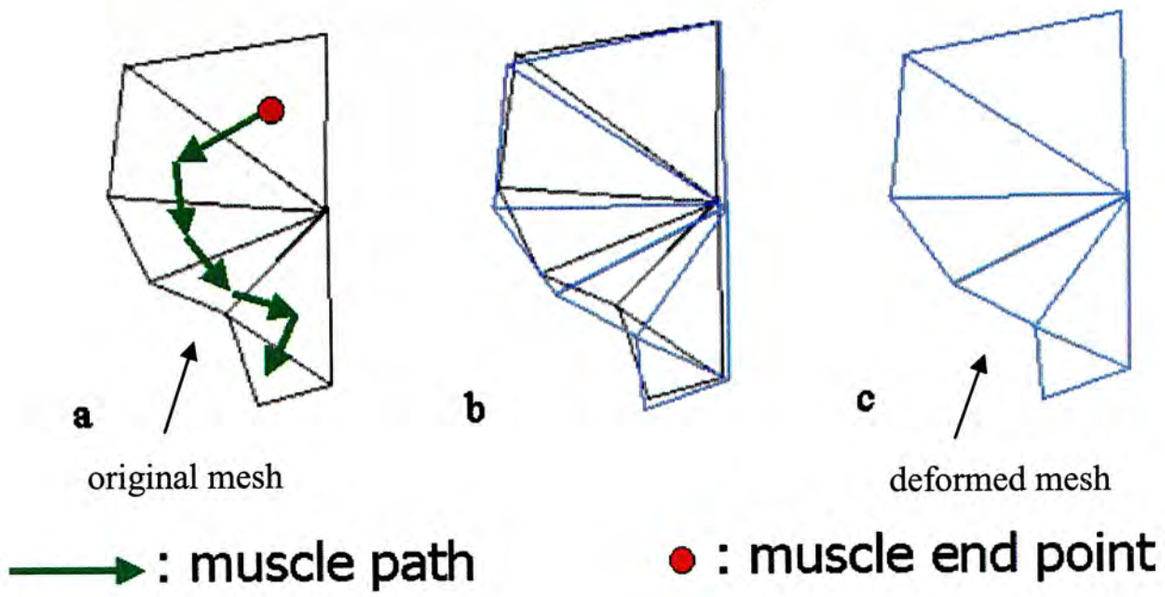


Figure 8.2. a) a facial muscle in normal state. b) overlapping of two facial muscles. c) a facial muscle after relaxation

By deforming the polygons of the facial muscles in this way, these polygons are ensured to follow the shape of the facial mesh.

8.3 The Muscle sliders

There are ten muscle sliders in the system, each of them controls the deformation of a facial muscle. The ten muscle sliders are shown in Figure 8.3. The range of a muscle slider is from –1.0 to 1.0, negative value denotes muscle contraction while positive value denotes muscle relaxation. The value of the muscle slider is used to control the percentage of the distance that should be moved between the two polygons centroids of a facial muscle.

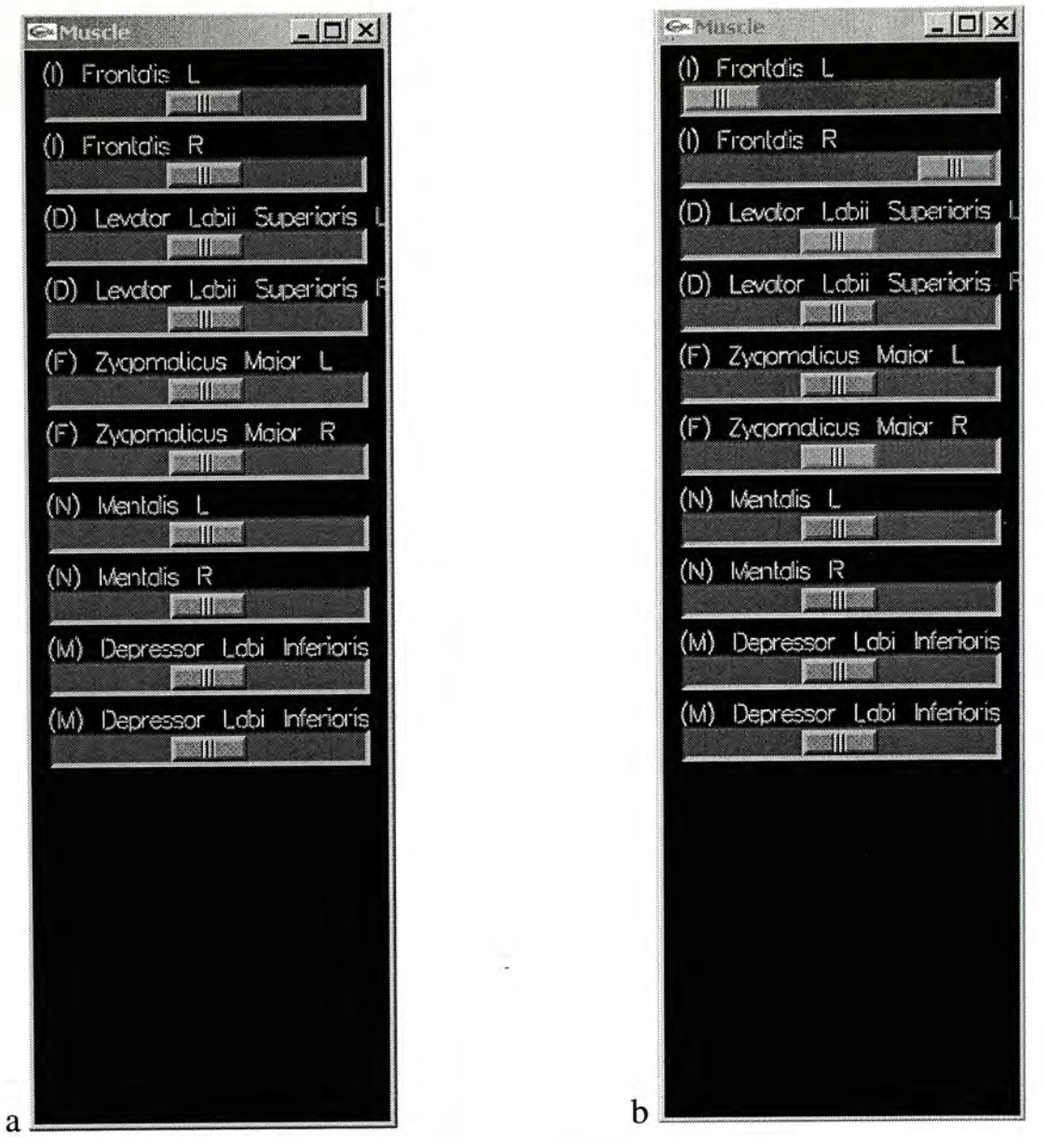


Figure 8.3. a) all ten muscle sliders in normal state.

b) the facial muscle Frontalis L is in contraction, the facial muscle Frontalis R is in relaxation

Chapter 9. Pre-computation

The system used one slider to control one facial muscle, moving the slider changes the boundary value of the corresponding facial muscle. A new solution of the system was then calculated by a matrix-vector multiplication which is a time consuming process. In order to improve the performance of the system the pre-computation and technique developed by James and Pai [22] are used to deform the facial model. By using the new update process, the process for computing a new solution drops from an $O(n^3)$ matrix-vector multiplication to O(mn) multiplication operation, where m is the number of polygons of the corresponding facial muscle and n is the number of polygons of the facial mesh.

9.1 Changing the Boundary Values

Deforming the facial muscles in the Facial Expression System means changing the boundary values of the adjustable displacement boundary conditions for the BEM engine. By using the initial reference solution, a new solution can be found by:

$$x_{new} = x_0 + \sum_{j \in S_u} (-K_0^{-1} H_j)\delta u_j + \sum_{j \in S_t} (-K_0^{-1} G_j)\delta t_j \qquad (9.1)$$

where x_0 and K_0 are the initial solution and initial stiffness matrix in equation (4.3), H and G are the initial traction kernel and displacement kernel in equation (4.2). δu_j and δt_j are the adjusted boundary values for the displacement and traction boundary conditions. S_u is the set which contains all the facial muscles and skull bone polygons, S_t is the set in which the elements are free to move in the Facial Expression System. They are mutually exclusive.

In the Facial Expression System the elements in S_u are always the same and free to move, therefore $\delta t_j = 0$ during the muscle movements. And equation (9.1) becomes

$$x_{new} = x_0 + \sum_{j \in S_u} (-K_0^{-1} H_j) \delta u_j \tag{9.2}$$

After solving the x_0 and K_0 in the BEM system, the terms $-K_0^{-1}H_j$ and are $-K_0^{-1}G_j$ will be computed and stored for the later update process.

Chapter 10. Implementation

The Facial Expression system is implemented in C++. The windowing tasks are handled by using the GLUT library [24]; the graphic rendering tasks are handled by using the OpenGL™ library; and the DevIL library [25] is used to handle the Jpeg (.jpg) image files. The program is divided into three modules, the first one implements a data structure to store the facial mesh, the second implements the Boundary Element Method, and the last one implements the facial modeling and facial feature recognition algorithms. The following sections discuss the structures of these three modules.

10.1 Data Structure for the facial mesh

The system uses a vertex array, an edge array and a polygon array to store the facial mesh. Figure 10.1 shows the data stored in each component. The edge information and adjacency information is stored for fast data retrieval during the facial modeling process.

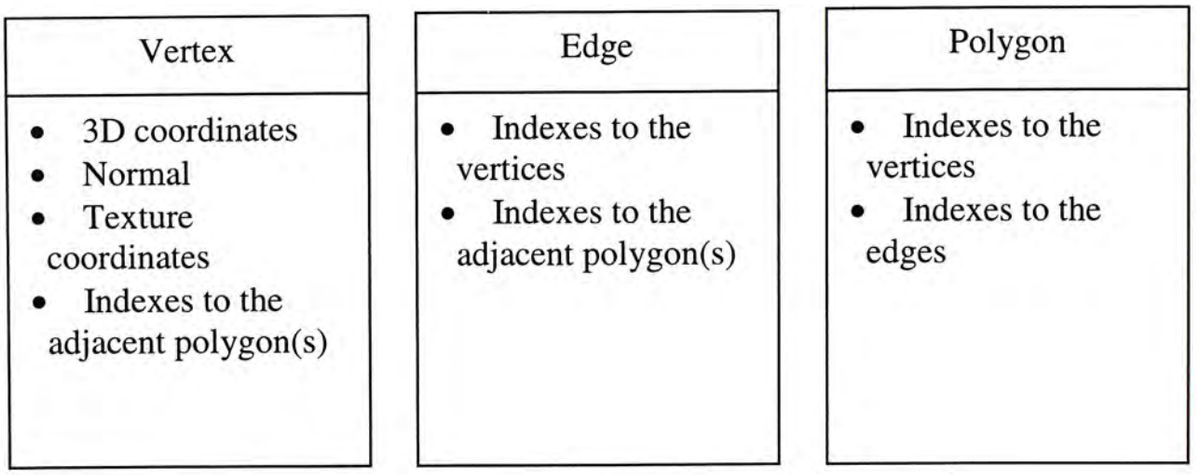


Figure 10.1 Data structure for the facial mesh

10.2 Implementation of the BEM engine

To setup the BEM engine, each polygon of the facial mesh is iterated. The system is then solved by using the Gauss Elimination.

10.3 Facial modeling and the facial feature recognition

The facial modeling and the facial feature recognition processes are implemented within different function body. To provide better control over the *Feature Boxes* and the facial muscles, a KBoundingBox class and a KMuscle class have been implemented. A *Feature Box* is an instance of the KBoundingBox class in the system. The ten facial muscles instanced from the KMuscle class are attached to the ten muscle sliders, each of the muscles is responsible for deforming itself.

Chapter 11. Results

This chapter shows some facial expressions produced by the Facial Expression System. The muscle sliders on the right hand side of these figures show the configuration of the facial muscles of a facial expression.

By using just ten facial muscles, several facial expressions can be produced by simply moving the muscle sliders.

In section 11.1, a low resolution man model is tested. Section 11.2 shows different facial expressions of a girl model with texture. Section 11.3 reproduces different facial expression of a man with the corresponding photo reference.

11.1 Example 1 (low polygon man face)

In this example, a low polygon model is used to test the system. The model consists of 660 polygons. Seven different facial expressions are produced in this example.

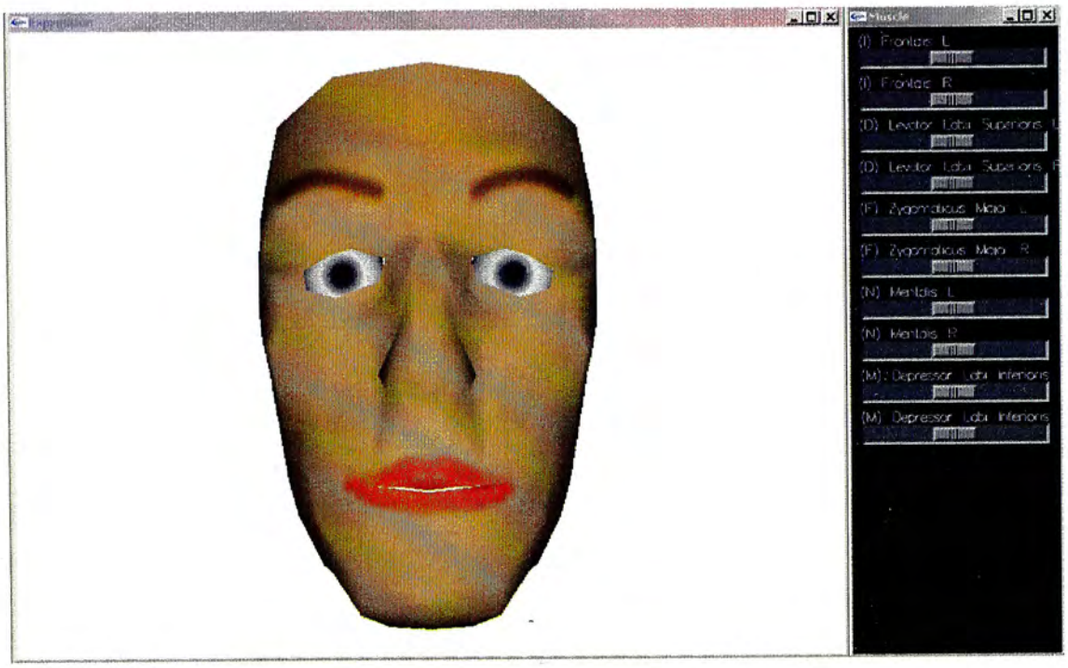


Figure 11.1. Normal state

Figure 11.2. Happy

Figure 11.3. Unhappy

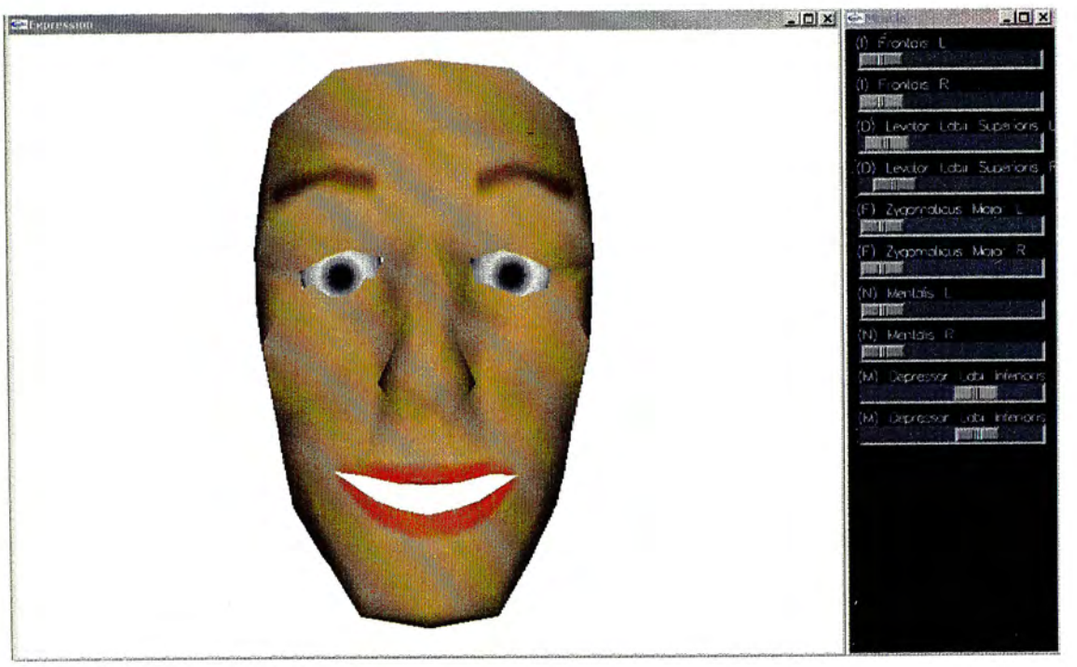


Figure 11.4. Laugh

Figure 11.5. Angry

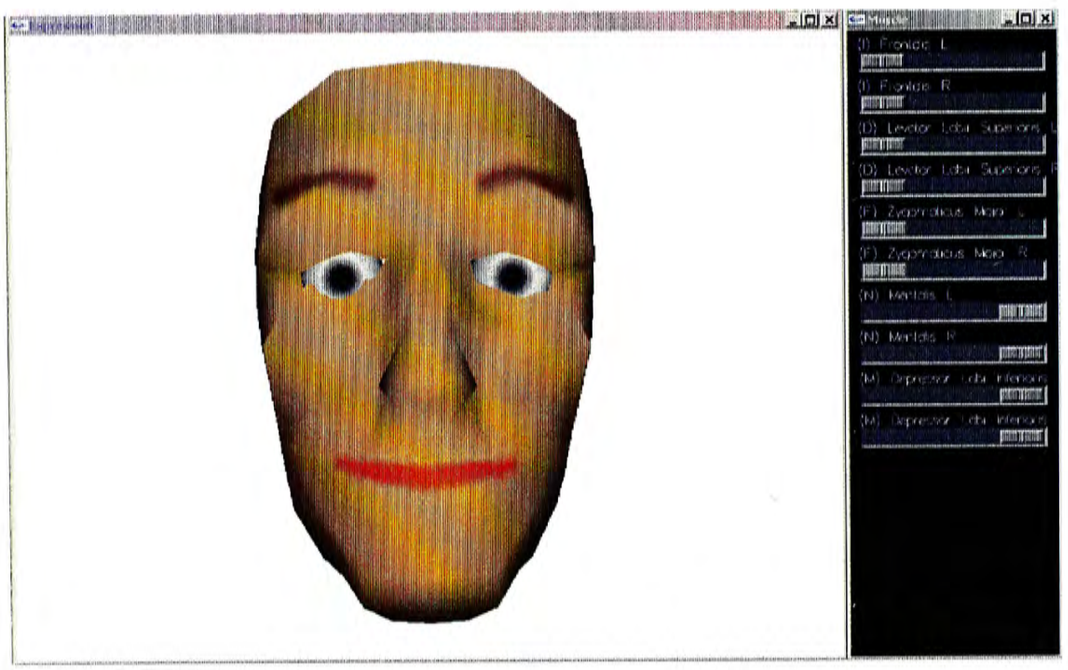


Figure 11.6. Satisfactory

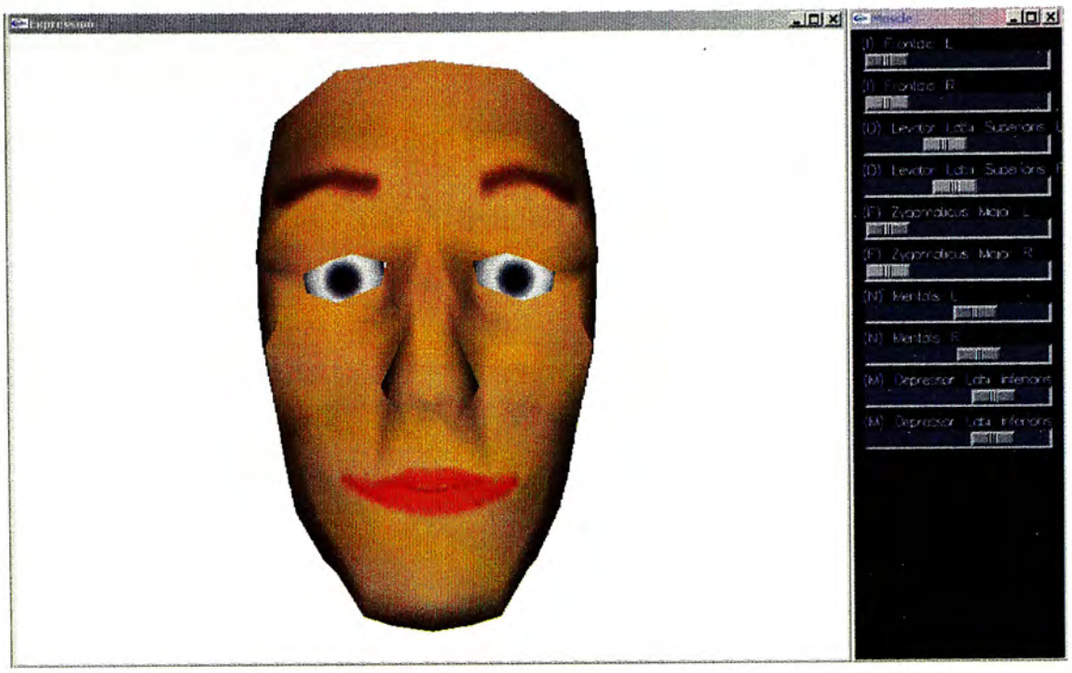


Figure 11.7. Smile

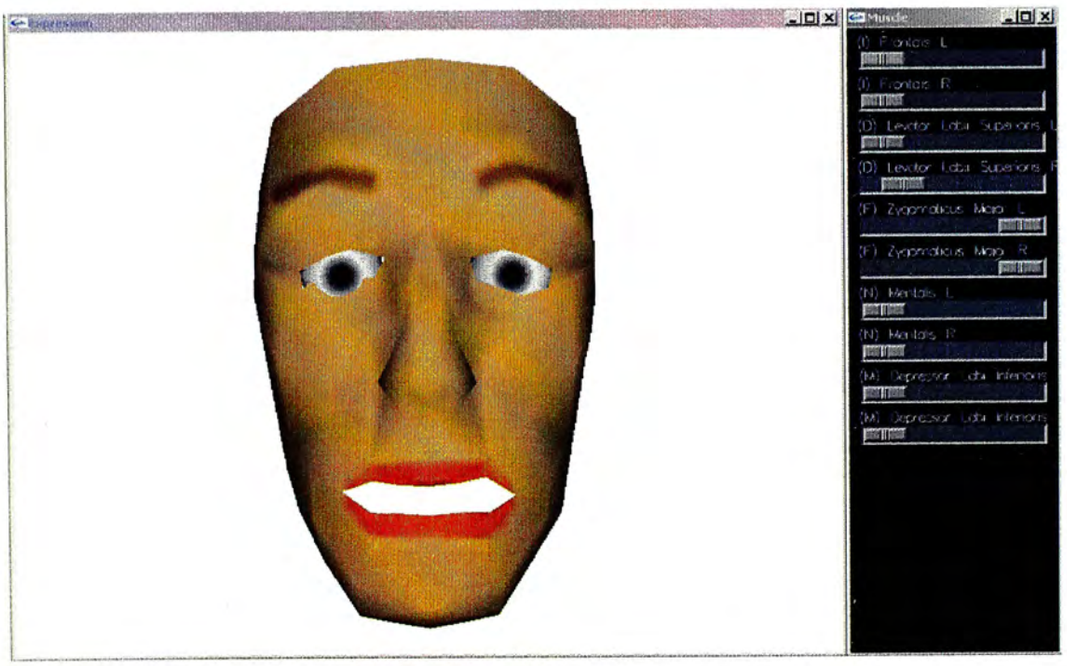


Figure 11.8. Surprise & disappointed

11.2 Example 2 (girl face)

In this example, a medium resolution girl model is used to test the system. The model consists of 1492 polygons. Six different facial expressions are produced in this example.

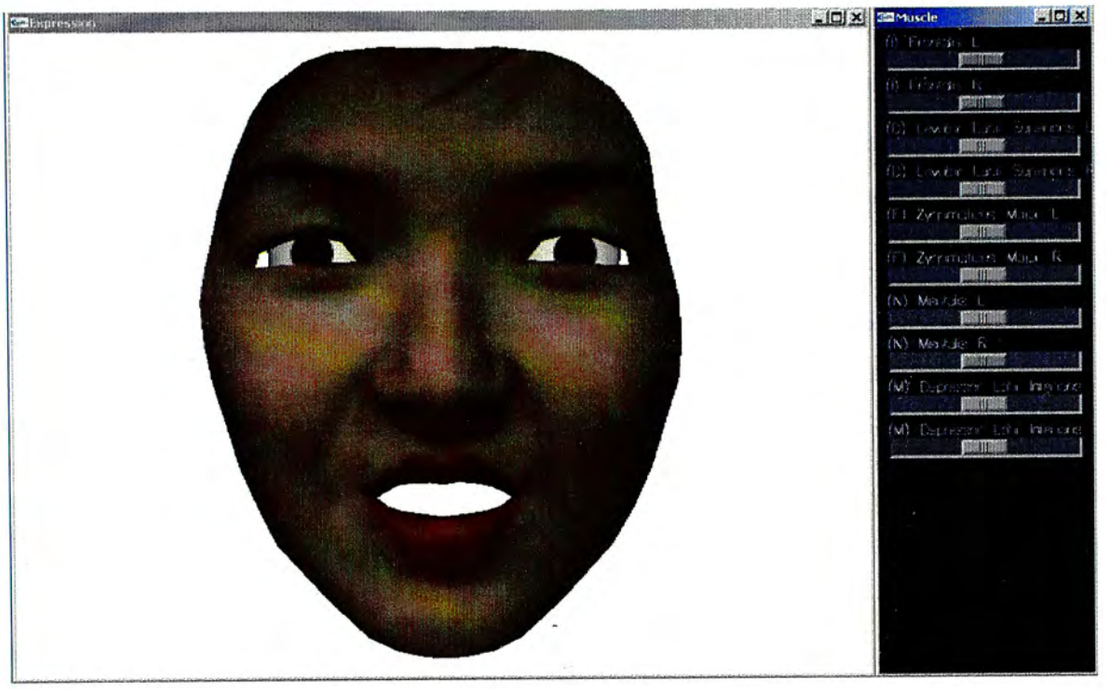


Figure 11.9 Normal state

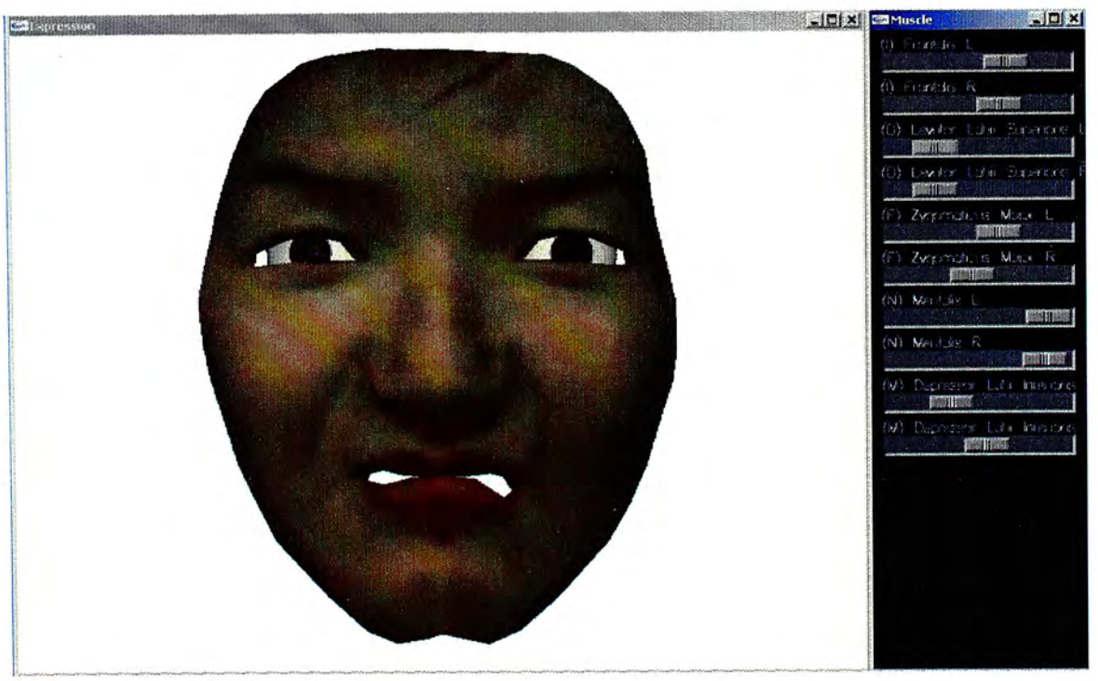


Figure 11.10 Angry

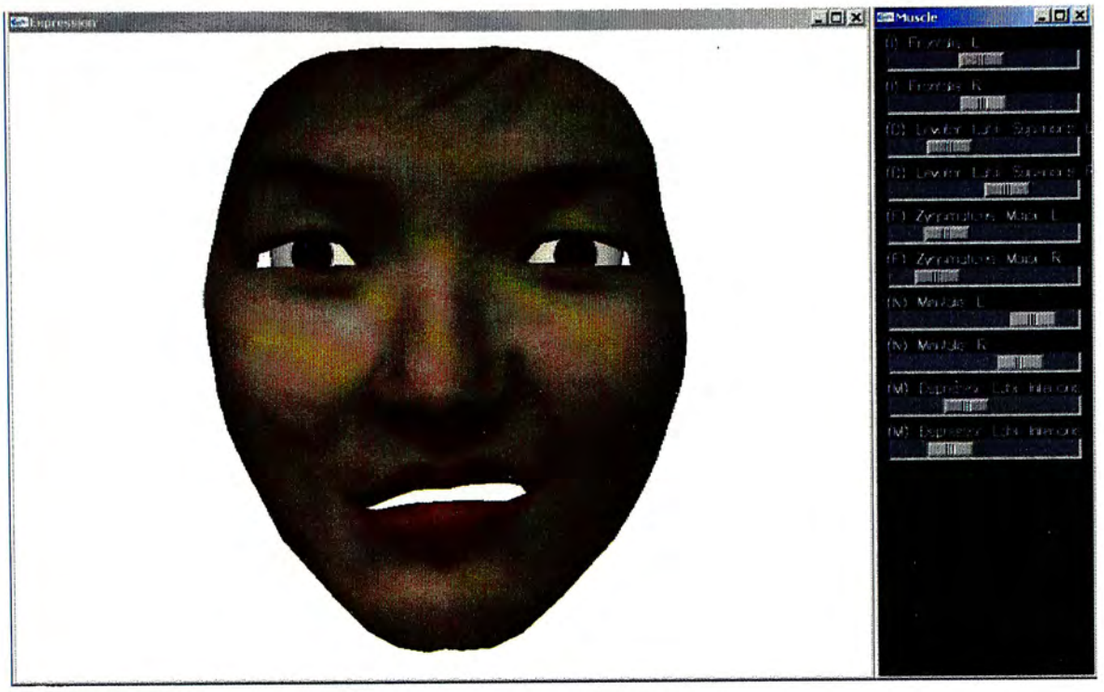


Figure 11.11 Fake smile

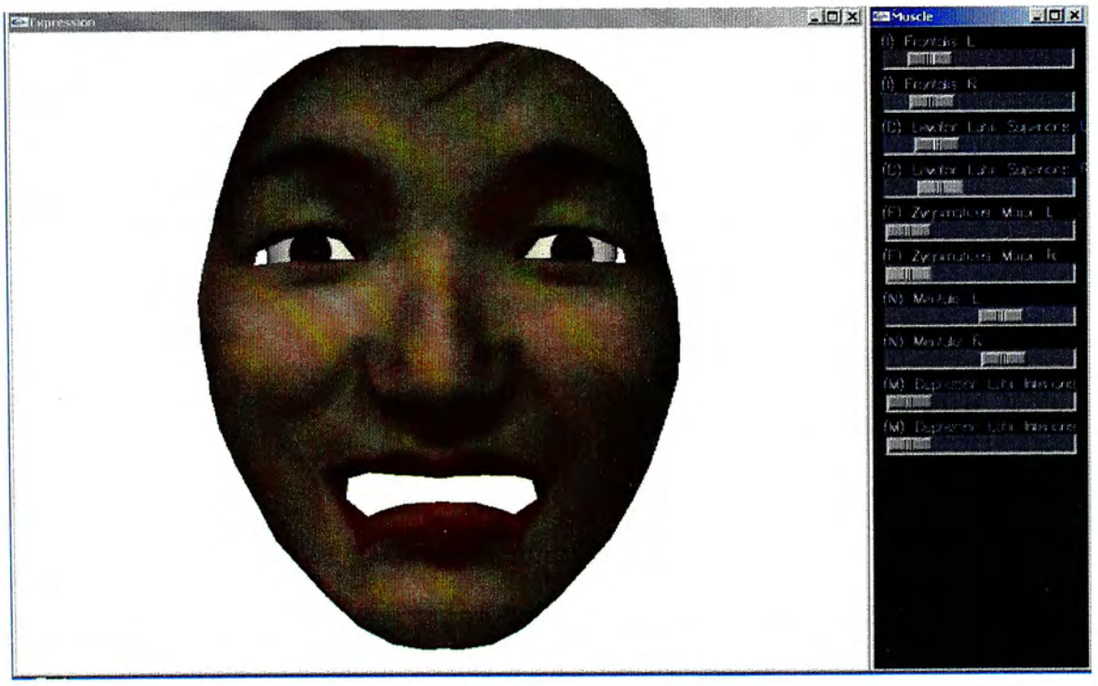


Figure 11.12 Frightened

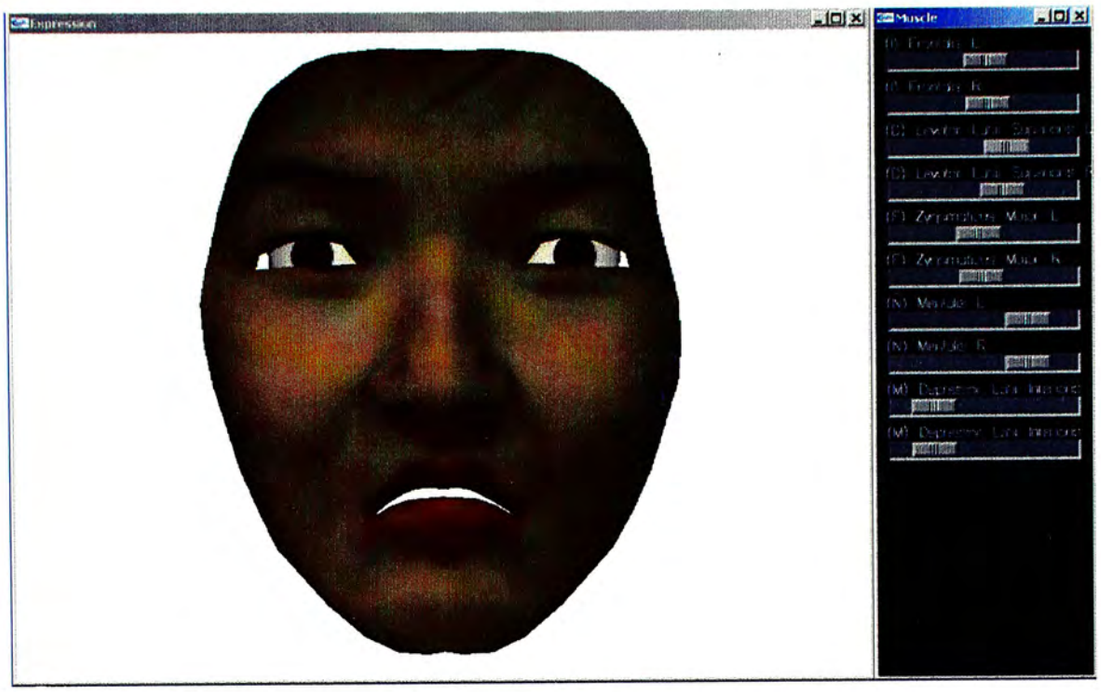


Figure 11.13 Unhappy

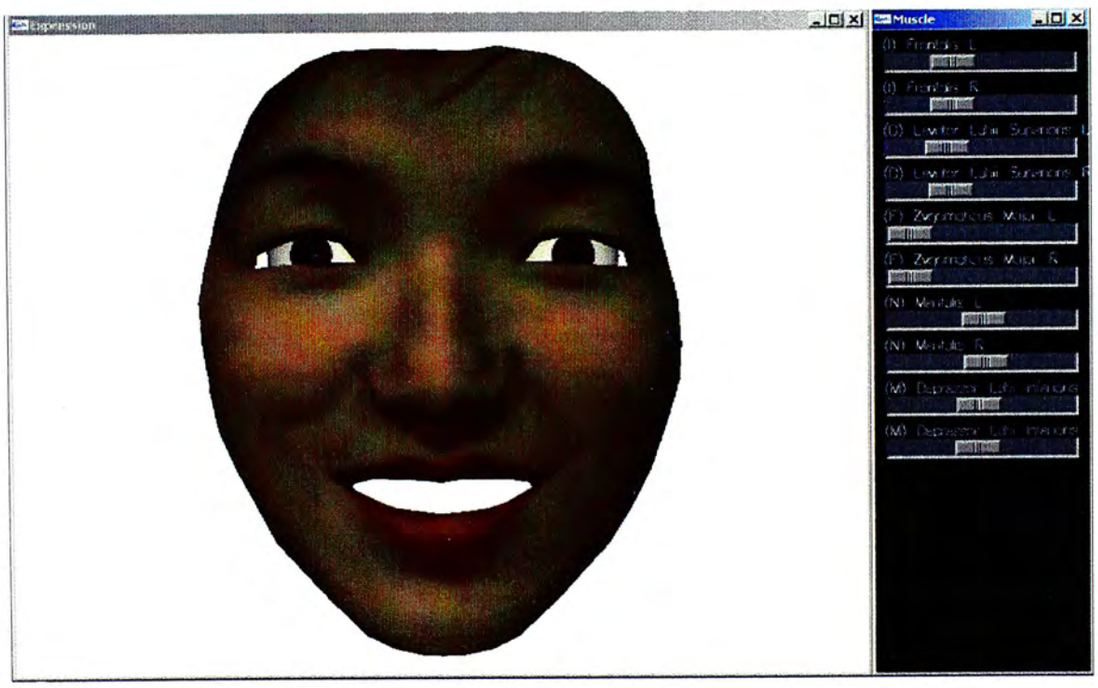


Figure 11.14 Happy

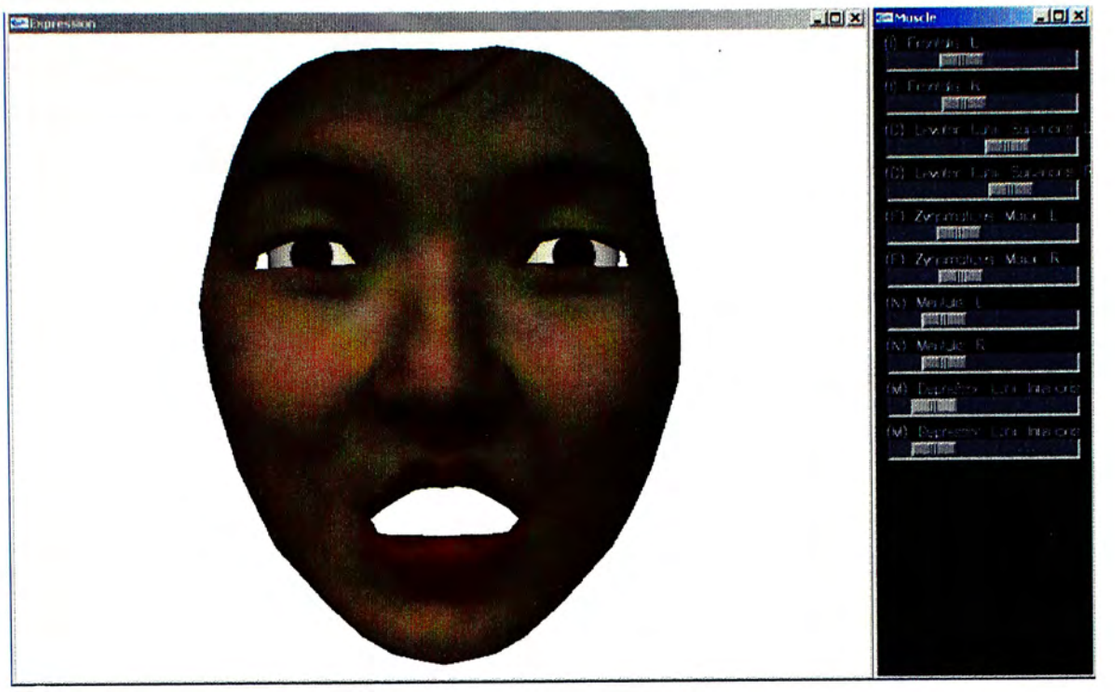
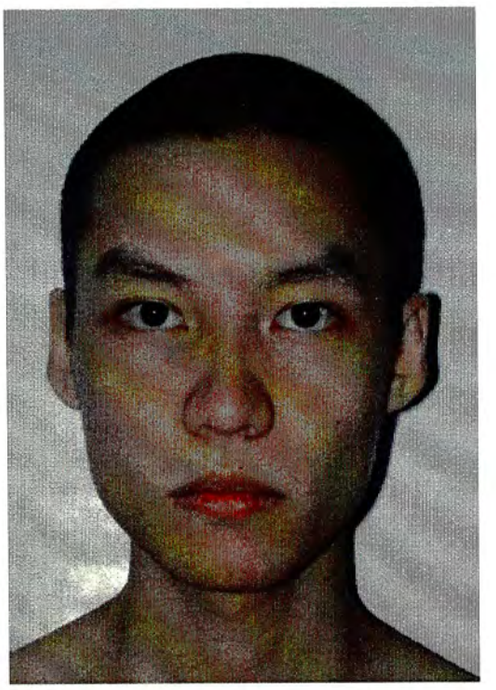


Figure 11.15 Surprised

11.3 Example 3 (man face)

In this example, a medium resolution man model is used to test the system. The model consists of 1524 polygons. Eight different facial expressions are produced in this example. The facial expressions produced by the system are compared with the corresponding real facial expressions produced by the author.



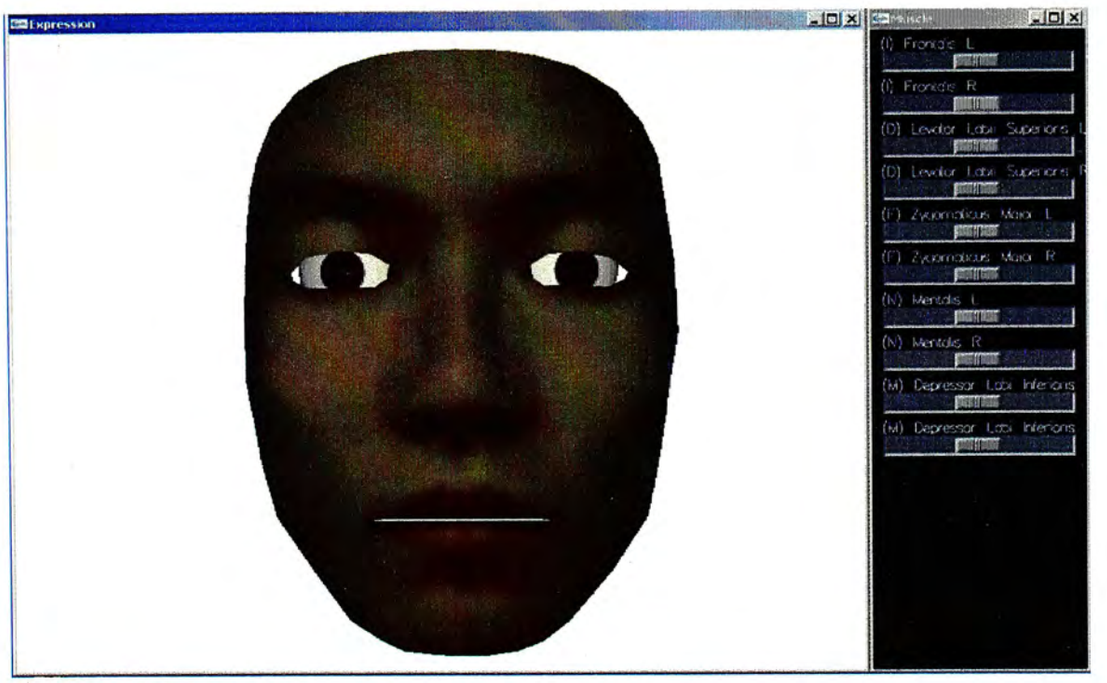


Figure 11.16 Normal state

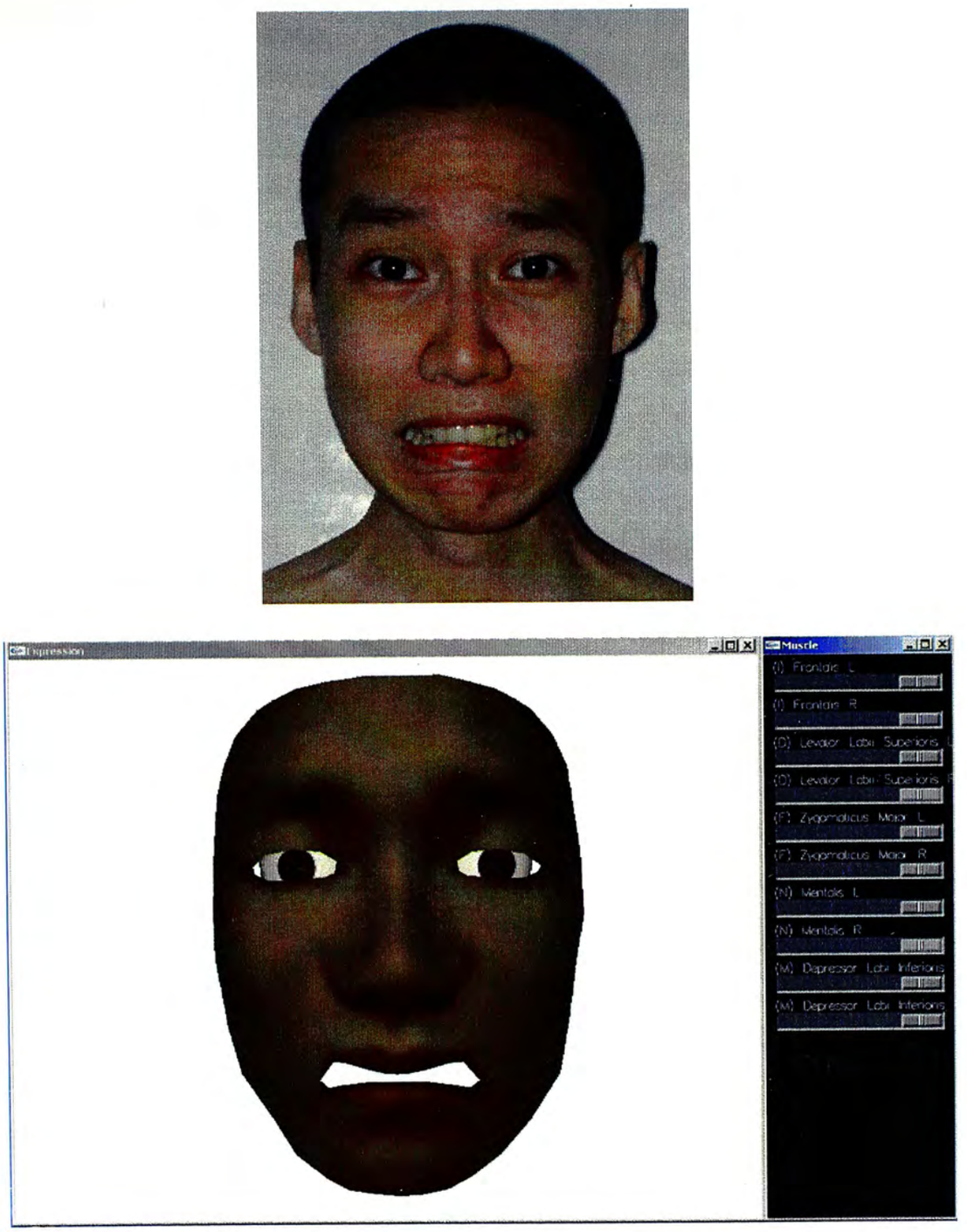
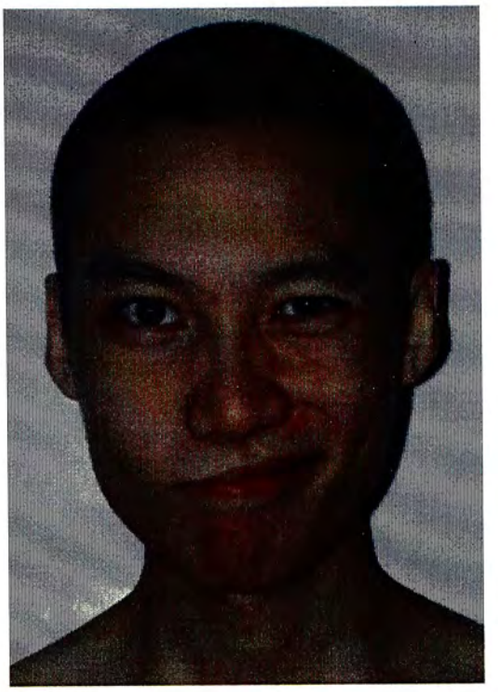


Figure 11.17 Frightened



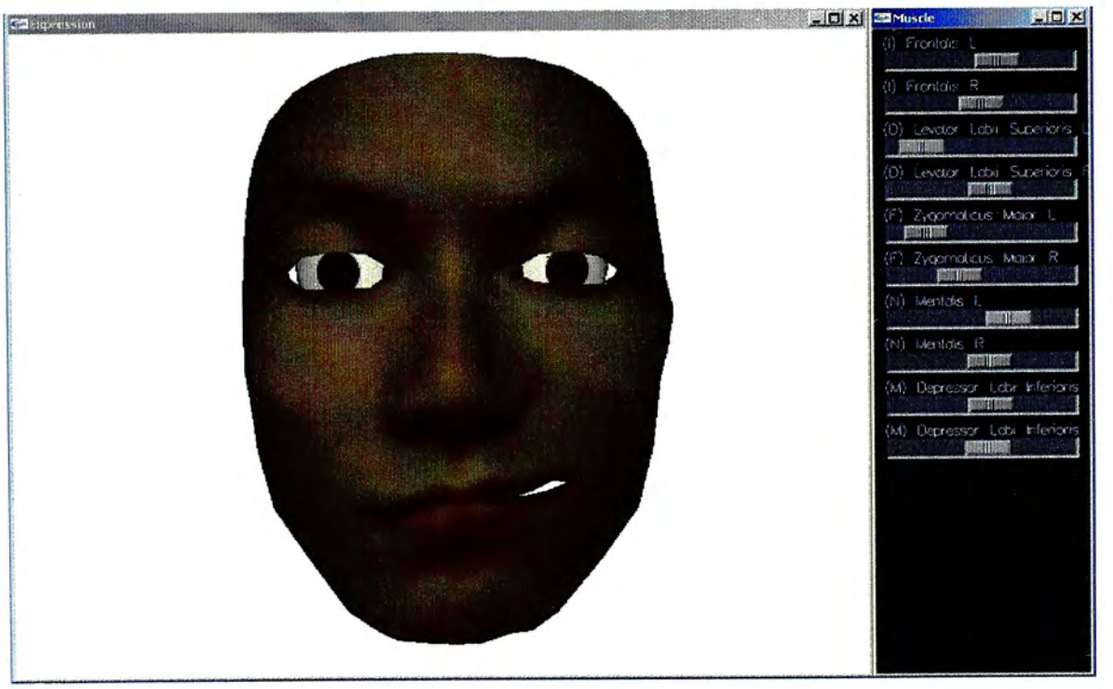


Figure 11.18 Fake smile

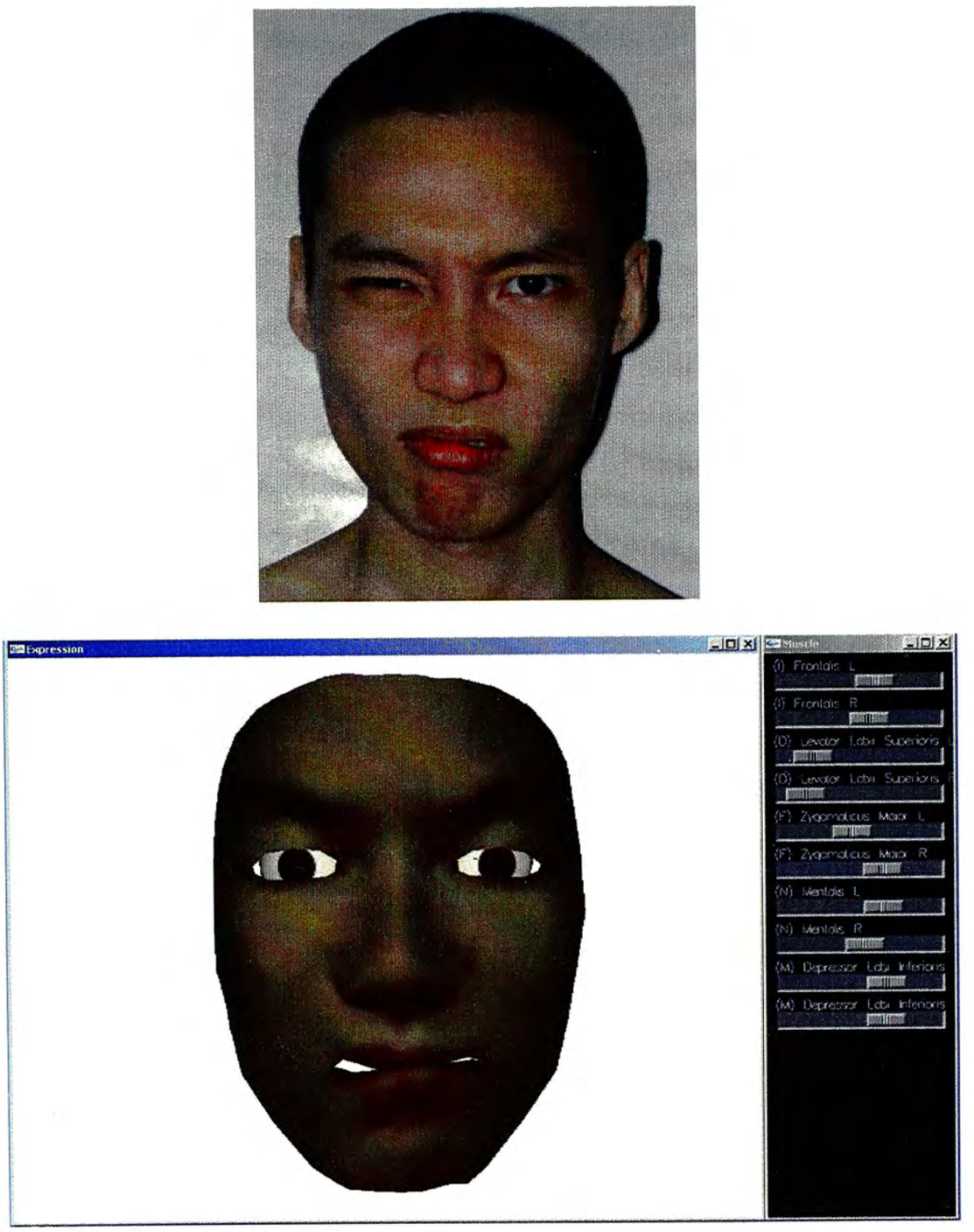
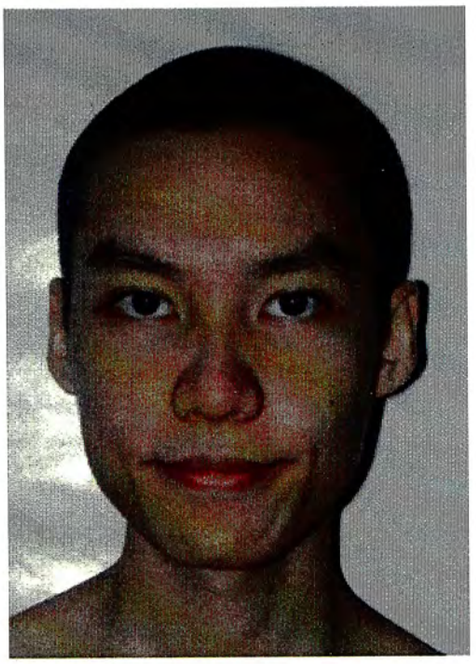


Figure 11.19 Angry



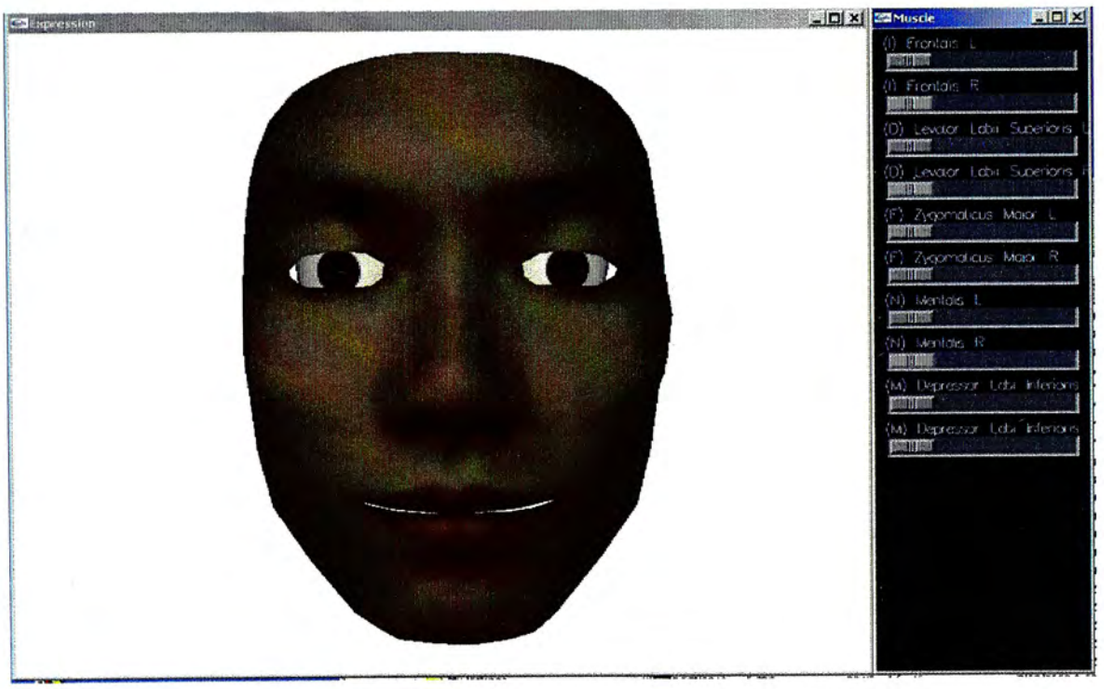


Figure 11.20 Smile

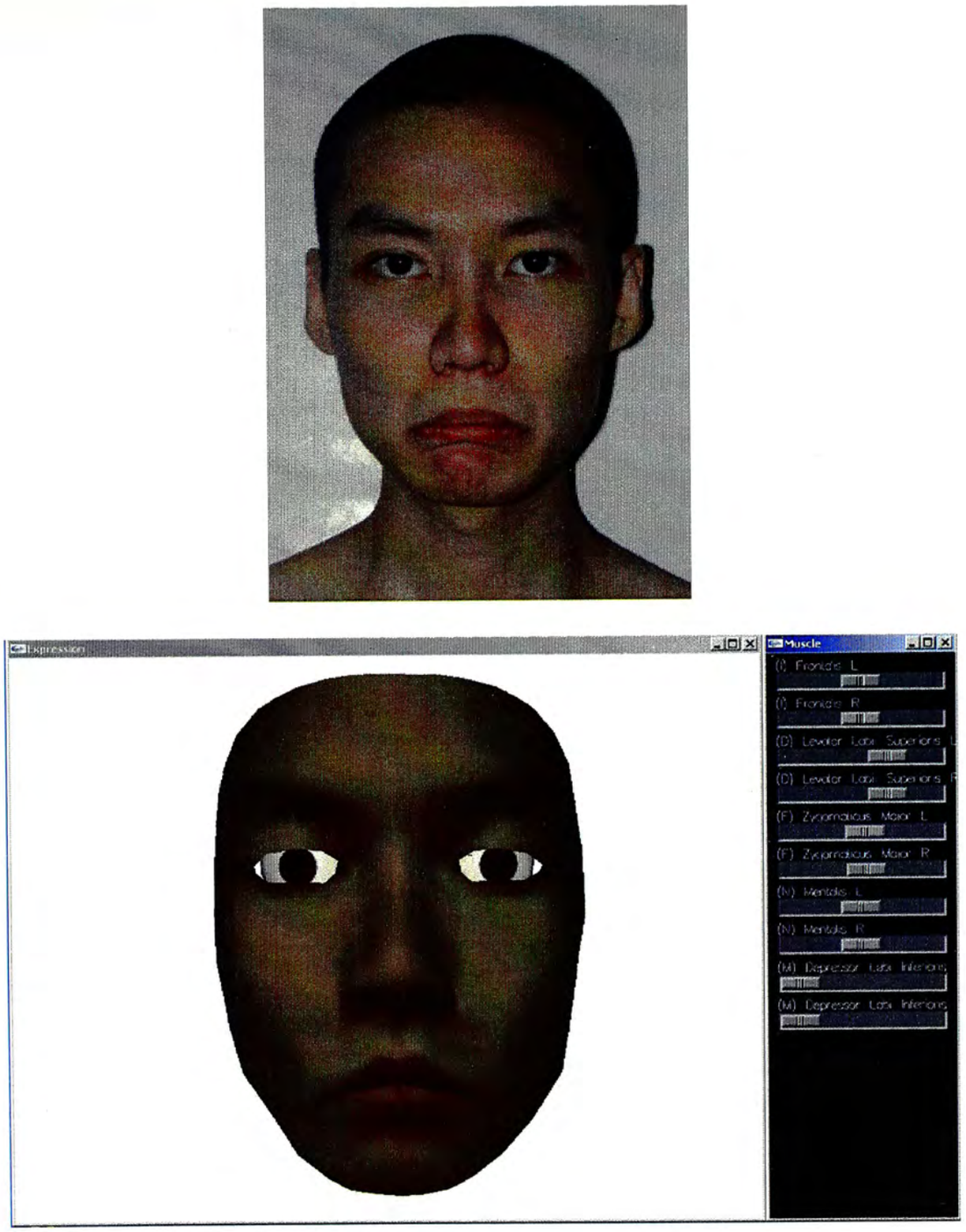


Figure 11.21 Unhappy

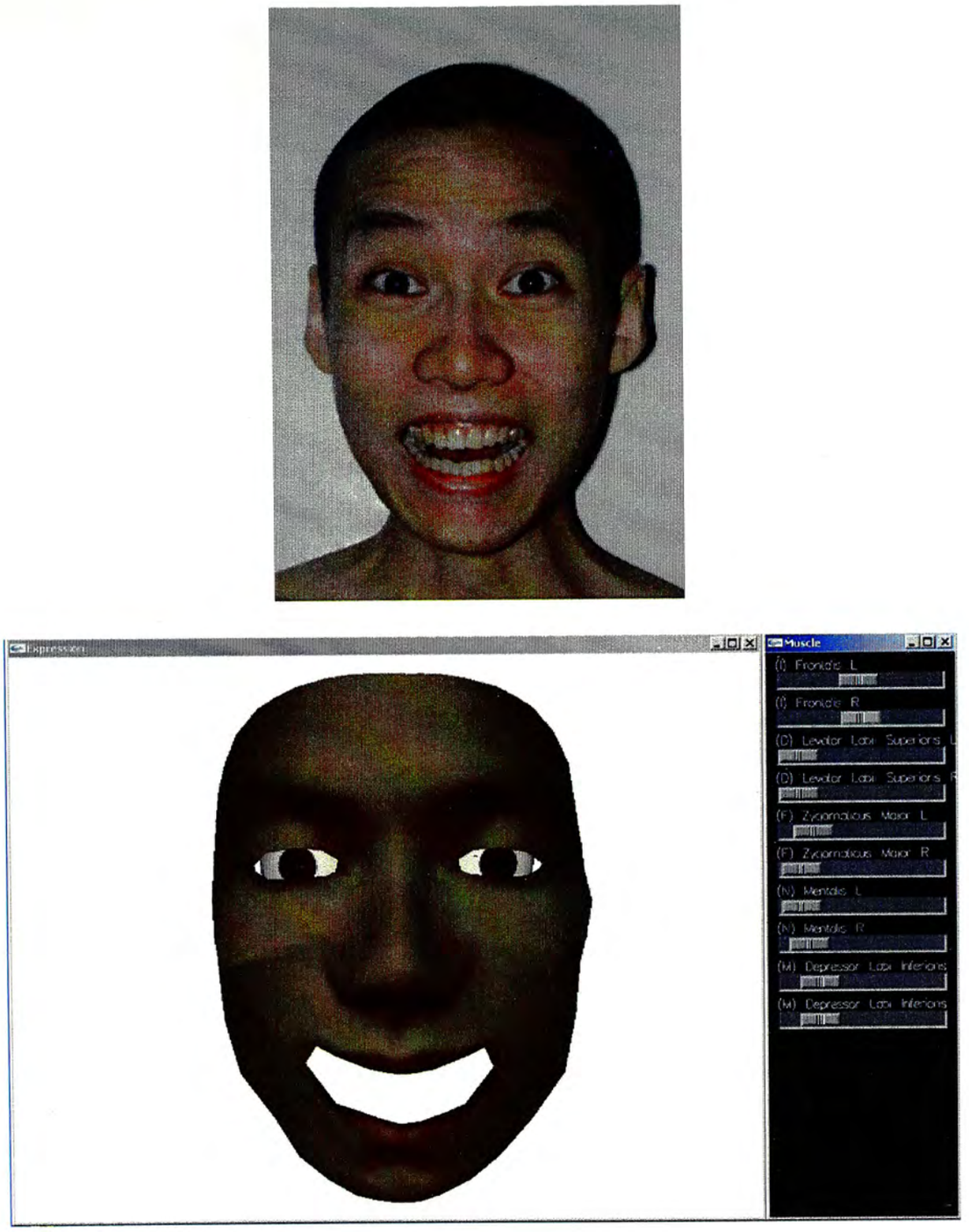


Figure 11.22 Very happy

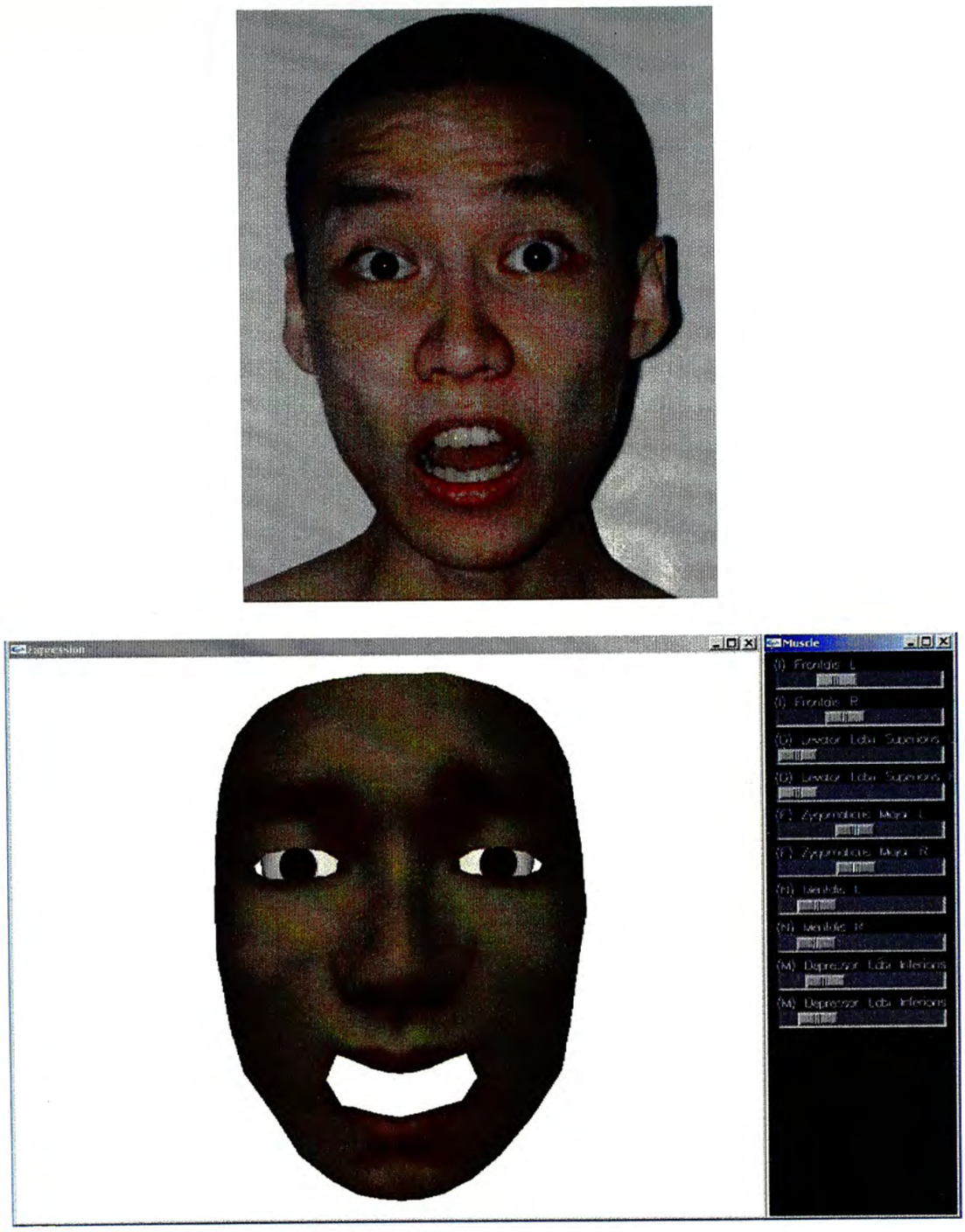
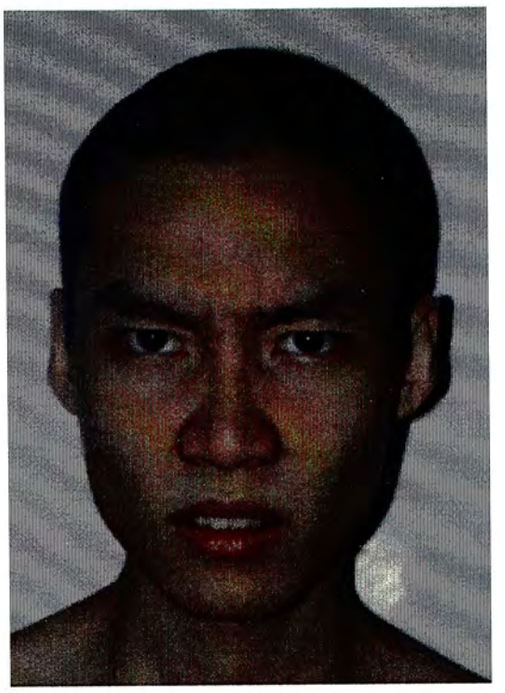


Figure 11.23 Surprised



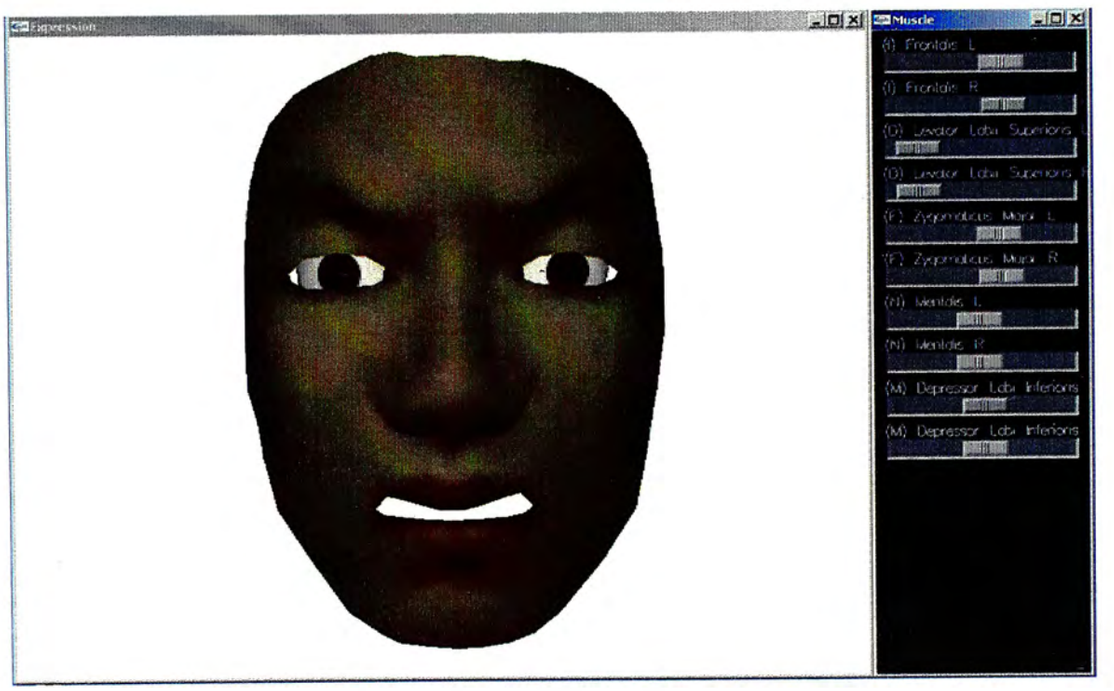


Figure 11.24 Hate

11.4 System evaluation

The strengths of the proposed system lie in three aspects. First, the facial feature and muscle recognition processes are automated, which highly enhances the efficiency of modeling process. Moreover, users can adjust the parameters found by the system, which allows flexibility of the sizes and positions of the facial features. Second, the Boundary Element Method is employed to govern the deformation of the facial mesh. The deformation results are physics based and accurate. Third, the system uses muscle sliders in producing a large number of facial expressions. The expressions produced are according to the human facial muscular profiles. Therefore, users can use the system to produce the expected expressions by referring the muscular profiles.

However, the system does have some limitations. The input mesh is required to be of a specified format, i.e. containing the eye, mouth and face contours. Despite that, the eye and mouth contours are usually common features of facial models while the face contour can be easily prepared by removing the hindbrain. Besides, the facial models are assumed to be linear elastic. A possible extension of this project is to apply other elastic models that can more accurately describe the facial tissues to the system.

Chapter 12. Conclusions

In this thesis, we have presented a new facial modeling and animation system based on the Boundary Element Method in linear elastic model. To the best of our knowledge, this is the first facial animation system that applies BEM deformation technique to facial expression modeling.

Our facial modeling and animation system contains some advantages over the existing ones. The facial templates preparation process in the technique using morphing of facial templates is relatively time consuming and requires human interactions whereas our system does not need such kind of process. When compared with those using non-physics muscle based deformation systems, our system can provide linear elastic deformation information that enhances deformation realism. Instead of using solid mesh as in FEM based facial animation system, our system employs BEM in deformation calculation which requires surface mesh only.

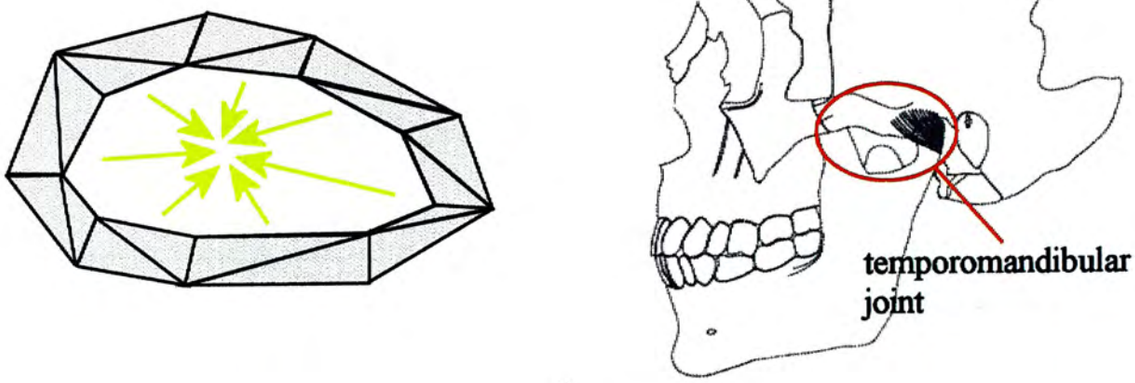
Besides, the facial feature and muscle recognition processes are completely automated, which reduces human interaction. By providing tools to resize the bounding box of the facial features, user can control the locations of the recognized facial muscles. To have better control, user can even edit the locations of the facial muscles and add/remove muscles manually.

The results show that different facial expressions can be efficiently produced by controlling the muscle sliders in the system.

However, as stated in Chapter 6, our feature recognition algorithm assumes that the input facial mesh has been axis aligned. One extension of this project is to revise the feature recognition algorithm to allow arbitrary orientation of the input facial mesh. Front face plane can be defined by the plane containing the centroids of

the three smaller contours (i.e. two eye contours and one mouth contour). Then, the facial mesh will be rotated by aligning the front face plane with the system x-y plane. To distinguish the eyes and the mouth, we can compare the sizes of the three contours. The two contours with similar sizes will be defined as the eye contours and the remaining one will be defined as the mouth contour. The eye level line is defined by connecting the centroids of the eye contours. Then, the facial mesh will be rotated by aligning the eye level line with the system x-axis. If the eyes are not in the positive y direction of the mouth, we rotate the facial mesh about the z-axis 180 degrees. Then, the facial mesh will be in a desired orientation.

Besides, the system does not consider the jaw, eyelid and eyeball motions, which limits the kinds of expression that it can produce. Specifically, the eyelid motion (eye close/open) is controlled by the Orbicularis Oculi muscles, which encircles the eye in concentric fibers. To produce simulation of orbicularis muscle contraction, we can define another type of muscles that are circular in shape and move concentrically as shown in Figure 12.1a. To simulate the jaw motion, we must define a jawbone on the facial mesh. The jawbone can be defined as the lower part of the new polygons which produced in the facial thickening process. By rotating the jawbone about the temporomandibular joints (Figure 12.1b), the system can produce mandible lowering and mouth opening/closing simulations. These are the future works of the system.



A b

Figure 12.1 a) polygons of the Orbicularis Oculi muscle connected into a circular loop. b) the temporomandibular joint

References

1. S. M. Platt, N. I. Badler. "Animating Facial Expressions". *Computer Graphics*, Vol. 15, No. 3, pages 245-252, Augest 1981.

2. F. I. Parke. "Parameterized Models for Facial Animation". *IEEE Computer Graphics and Applications*, pages 61-68, 1982.

3. K. Waters. "A Muscle Model for Animating Three-Dimensional Facial Expression". *Computer Graphics*, Vol. 21, No. 4, pages 17-24, July 1987.

4. JP. Gourret, N. Magnenat Thalmann, D. Thalmann. "Simulation of Object and Human Skin Deformations in a Grasping Task". *Computer Graphics*, Vol. 23, No. 3, pages 21-30, July 1989.

5. G. Celniker and D. Gossard. "Deformable Curve and Surface Finite-Elements for Free-Form Shape Design". *Computer Graphics*, Vol. 25, No. 4, pages 257-266, July 1991.

6. D. T. Chen and D. Zelter. "Pump It Up: Computer Animation of a Biomechan- ically Based Model of Muscle Using the Finite Element Method". *Computer Graphics*, Vol. 26, No. 2, pages 89-98, July 1992.

7. Y. C. Lee, D. Terzopoulos, and K. Waters. "Constructing Physics-Based Facial Models of Individuals". *Graphics Interface ' 1993*, pages 1-8, 1993.

8. H. Delingette, G. Subsol, S. Cotin, J. Pignon. "A Craniofacial Surgery Simulation Testbed". P*roceedings of the Visualization for Biomedical Computing (VBC'94)*, Oct. 1994.

9. Y. C. Lee, D. Terzopoulos, and K. Waters. "Realistic Modeling for Facial Animation". *SIGGRAPH '1995*, pages 55-62.

10. M. Bro-Nielsen and S. Cotin. "Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation". *EUROGRAPHICS, 1996*, Vol. 15, No. 3, pages C57-C66, 1996.

11. E. Keeve, S. Girod, P. Pfei, B Girod. "Anatomy-Based Facial Tissue Modeling Using the Finite Element Method". *IEEE Visualization '1996, Proceedings*, pages 21-28, 1996.

12. R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser, Y. I. H. Parish. "Simulating Facial Surgery Using Finite element Models". *SIGGRAPH '1996*, pages 421-428.

13. F. Scheepers, R. E. Parent, W. E. Carlson, S. F. May. "Anatomy-based Modeling of human Musculature". *SIGGRAPH '1997*, pages 163-172.

14. J. Wilhelms and A. Van Gelder. "Anatomically Based Modeling" *SIGGRAPH '1997*, pages173-180.

15. Q. H. Zhu, Y. Chen, and A. Kaufman. "Real-time Biomechanically-based Muscle Volume Deformation using FEM". *EUROGRAPHICS '1998*, Vol. 17, No. 3, pages C276-C284, 1998.

16. S. H. M. Roth, M. H. Gross, S. Turello, F. R. Carls. "A Bernstein-Bézier Based Approach to Soft Tissue Simulation". *EUROGRAPHICS '1998*, Vol. 17, No. 3, pages C285-C294, 1998.

17. R. M. Koch, M. H. Gross, A. A. Bosshard. "Emotion Editing using Finite Elements". *EUROGRAPHICS '1998*, Vol. 17, No. 3, pages C295-C302, 1998.

18. S. Cotin, H. Delingette, and N. Ayache. "Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation". *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 1, pages 62-73, 1999.

19. P. Ekman and W. V. Friesen. "Manual for the Facial Action Coding System". *Consulting Psychologists Press*, Palo Alto, CA. 1978.

20. H. Delingette. "Simplex meshes: a general representation for 3d shape reconstruction". *Technical Report 2214, INRIA*, March 1994.

21. National Library of Medicine. "The Visible Human Project". *http://www.nlm.nih.gov/extramural research.dir/visible human.html*, 1995.

22. D. L. James and D. K. Pai. "ARTDEFO: Accurate Real Time Deformable Objects", In *Computer Graphics (SIGGRAPH 99 Conference Proceedings)*, 1999

23. K. C. Hui, H. C. Leung. "Virtual Sculpting and Deformable Volume Modelling", *Proceedings of Information Visualization*, pages. 664-669, 2002.

24. OpenGL Utility Toolkit. "GLUT" *http://www.pobox.com/~nate/glut.html.*

25. Developer's Image Library. "DevIL" *http://www.imagelib.org.*

26. F. I. Parke and K. Waters. *Computer Facial Animation.* A K Peters, Ltd., 1996.