Improvements on a simple muscle-based 3D face for realistic facial expressions

The Duy Bui Dirk Heylen Anton Nijholt
University of Twente
Department of Computer Science
The Netherlands
{theduy,heylen,anijholt}@cs.utwente.nl

Abstract

Facial expressions play an important role in face-to-face communication. With the development of personal computers capable of rendering high quality graphics, computer facial animation has produced more and more realistic facial expressions to enrich human-computer communication. In this paper, we present a simple muscle-based 3D face model that can produce realistic facial expressions in realtime. We extend Waters' muscle model to generate bulges and wrinkles and to improve the combination of multiple muscle actions. In addition, we present techniques to reduce the computation burden on the muscle model.

1 Introduction

Facial expression plays an important role in face-to-face communication [1]. With the development of personal computer capable of rendering high quality graphics, computer facial animation has produced more and more realistic facial expressions that can be used to enrich human-computer communication.

Approaches to facial animation range from simple ones to achieve realtime animation ([14] to more sophisticated ones to achieve photo quality expressions ([9], [5]). The methods can be divided into two branches: geometry and image manipulation. We will only be concerned with the former. Geometry manipulation comprises several techniques including: interpolation ([12]), parameterizations ([4]), finite element methods ([8], [9]), pseudo muscle models ([6]) and physics based muscle models ([5], [15], [16], [18]). For a detailed survey see [10].

Interpolation techniques require the storage of different facial positions while parameterizations lose the generality when applied to a new facial topology. Heavy physics based models like [8], [9] and [16], produce realistic facial expression by modelling the detailed anatomical structure and dynamics of the human face. However, because massive computation is required, these models are not used widely for realtime animation. Pseudo muscle models just ignore the complicated underlying anatomy and deform only the thin facial mesh. These models normally fail to provide a generic way of producing bulges and wrinkles in the skin. Instead, special rendering techniques like bump mapping are used to produce wrinkles in specific regions. Moreover, not much attention is paid to the interaction of multiple muscles.

In this paper, we present a simple muscle-based 3D face model that can produce realistic facial expressions in realtime. We extend Waters' muscle model [18] to generate bulges and wrinkles and to improve the combination of multiple muscle actions. In addition, we present techniques to reduce the computation burden on the muscle model.

Section 2 presents the structure of the face model. The muscles that control the facial animation are discussed in Section 3. Section 4 describes how the bulges and wrinkles are generated in this model. The techniques to increase the animation speed are described in Section 5.

2 The face model

Primarily, we use triangular polygons to represent the face. We have it rendered and animated with OpenGL technologies. The polygonal structure makes the face easy to be deformed by a muscle model. Moreover the polygonal structure allows fast rendering by current regular personal computers. In addition, we use a simple B-spline surface to model more realistic lips involved in facial expression and visemes generation during speech.

2.1 The face mesh

Our face mesh data was obtained from a 3D scanner. We processed the data to improve the animation performance while keeping the high quality of the model. This process contains two phases.

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE

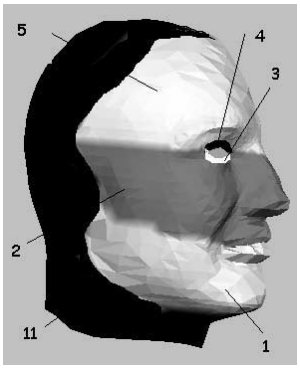


Figure 1. The region division on right half of the face

In the first phase, we reduced the number of vertices and polygons in some parts of the face. We took the idea from Greta [14] to have more vertices and polygons in the expressive parts. These are the regions around the eyes, the nose, the mouth and the forehead. This approach still keeps the great level of detail in these expressive parts that are dedicated to send communication signals and emotional expressions. The reduction of polygons in other parts increases the animation speed. Our final 3D face model contains only 2480 vertices and 4744 polygons (Greta contains around 15000 polygons) and is shown in Figure 13.

In the second phase, we divided the face into regions. Region division has been used in Greta to limit and control the displacement of polygon vertices induced by facial animation parameters (FAPs). Similarly, we use that technique to control the displacement of polygon vertices produced by muscle contraction. Moreover, we use region division to improve the muscle model, which will be discuss in Section 3.1.

Based on the distribution of muscles, we divide the face into eleven regions: right lower face (1), right middle face (2), right lower eyelid (3), right upper eyelid (4), right upper face (5), left lower face (6), left middle face (7), left lower eyelid (8), left upper eyelid (9), left upper face (10), and the non-animated region consisting of the rest of the head (11). The regions on the right half of the face can be seen in Figure 1, while the regions on the left half of the face are in similar positions. This division keeps it easy for us to find which regions a muscle has effect on while trying to maximize the number of regions on the face.

There are also some sub-regions inside those regions above. They are: the lower lip which is inside the lower face regions (for both left and right side), the upper lip which is inside the the middle face regions (for both left and right side), and the eyebrows which are inside the upper face regions.

We reordered the vertices of the face mesh in the data file by regions. We introduce a simple description file associated with the data file which looks like:

Lower face start vertex 0 Lower face end vertex 159

Middle face start vertex 160 Middle face end vertex 484 ...

The division in regions makes it easy for rendering special parts of the face like lips and eyebrows. It also helps to prevent visual artifacts generated by the displacement of vertices in the regions that are not affected by a muscle's contraction. For example, without specifying the eyelid regions, the polygons in the eyelids can be distorted by **Frontalis** muscles as the eyelids may lie in their zone of influence, which results in unnatural animation. Finally, the region division enables improvements on the muscle model as we will show later.

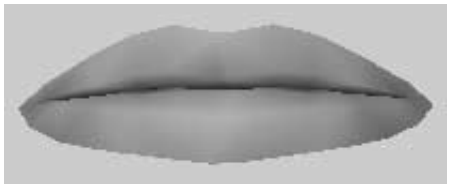


Figure 2. The lips

2.2 The lip model

The lips are the most mobile part of the face. They are also a very expressive part of the face during facial expressions and they participate in the articulation of speech. We will not discuss the viseme generation for speech in this paper.

Because of their importance, we paid extra attention to the lips. We based the lip model on the proposal in [7]. Our lip model is a B-spline surface with 24×6 control grid. We use a B-spline surface to ensure the smoothness of the lips after distortion by the muscle model. The lips are shown in Figure 2.

2.3 The eyeball model

To enable gaze behavior, we have implemented the eye tracking algorithm proposed in [11]. Note that an eye can only move in a range constrained by the eye hole, which has an ellipse shape. We have added this constraint to the algorithm so the eyes cannot rotate to impossible positions. The eye movement is independent of the facial muscle movements.

3 The muscles that drive the facial animation

In this section, we will describe the muscles that control the animation on the 3D face. The muscles are mainly based

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE

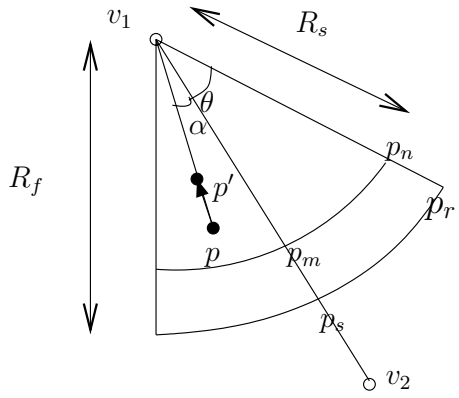


Figure 3. The linear muscle model

on Waters' muscle model [13], [18]. Waters has modelled three types of muscle: the vector muscles used for the majority of facial muscles, the sheet muscle for the **Frontalis**, and the sphincter muscle for the **Orbicularis Oris**.

We have extended the vector muscle model to improve the combination of multiple muscle actions (Section 3.1). For the **Frontalis**, we use a set of vector muscles instead of the sheet muscle because the forehead is not completely flat. The mouth and eye muscles (**Orbicularis Oris** and **Orbicularis Oculi**) are rather complex. Modelling these muscles by Waters' sphincter muscle does not give realistic results. We therefore implement these muscles as pseudo muscles, which directly manipulate the 3D mesh to produce similar effects as the real muscles do (Section 3.2 and 3.3). We also implement a realistic jaw rotation, which is presented in Section 3.4.

3.1 The vector muscles

We use the vector muscle model from [18]. The muscle's zone of influence are illustrated in Figure 3. The muscle is modelled as a vector from v_2 to v_1. R_s and R_f represent the fall-off radius start and finish respectively. The new vertex p' of an arbitrary vertex p located on the mesh within the segment $v_1 p_r p_s$, along the vector (p, v_1), is computed as follows:

$p' = p + cos(\alpha) k r \frac{pv_1}{||pv_1||}$

where α is the angle between the vector (v_1, v_2) and (v_1, p), D is $||v_1 - p||$, k is a fixed constant representing the elasticity of skin, and r is the radial displacement parameter:

$$r = \begin{cases} cos(1 - \frac{D}{R_s}) & \text{for p inside sector } (v_1 p_n p_m) \\ cos(\frac{D - R_s}{R_f - R_s}) & \text{for p inside sector } (p_n p_r p_s p_m) \end{cases}$$

The problem associated with the model arises when a mesh vertex is under the influence of multiple muscle actions. Waters combined these muscle actions by applying the displacements caused by them on a vertex one by one. When the vertex is shifted out of the zone of influence of adjoining muscle vectors and the contractions become isometric, this approach produces unnatural results, which has been discussed by Wang [17]. Wang took another approach by summing up the displacement and then apply this to the vertex. This approach itself also generates strange results as in Figure 4(d) and 5(b). Both Waters and Wang also tried to eliminate unnatural results by truncating the displacement of a vertex by the vertex maximum displaced distance. This process requires that the maximum displacement of each individual vertex is determined, which is difficult. Assigning this distance as the maximum of the displacements generated by each single muscle maximum contraction will not eliminate the problem completely as the vertex is still shifted out of the muscles' zone of influence. Assigning this distance as the minimum of the displacements generated by each single muscle's maximum contraction will incorrectly truncate the displacements when a single muscle contracts.

We combine muscle contractions by simulating their parallelism. For a vertex inside multiple muscles' zone of influence, we interactively apply small units of contraction levels to the vertex until there is no more contraction to apply. Each time a small unit δ_c of contraction levels is applied, the displacements of the vertex (caused by the muscles that have the vertex in their zone of influence) are summed up and applied. Our approach with $\delta_c = 0.2$ (the maximum value of a muscle contraction level is 1.0) produces results as in Figure 4(c) and 5(a). This approach is also used to combine the contractions of other muscle types.

3.2 Orbicularis Oris

Physiologically, the **Orbicularis Oris** is not just a simple sphincter muscle but a combination of muscles that can drive the mouth in different directions [2]. Our pseudo muscle implementation for the **Orbicularis Oris** is adopted from [7].

Recall from Section 2.2, the lip is a B-spline surface with a 24×6 control grid. In our model, the **Orbicularis Oris** affects only the lip surface. The surface is deformed by displacing the control points. The displacement of a control point p_i due to the contraction of the muscle **Orbicularis Oris** is described as:

$p'_i = o(\theta_i + e(p_i) + x_i)$

where o is the contraction level of the **Orbicularis Oris** muscle, θ is the maximum rotation due to the puckering of the lips, and x_i is the maximum extrusion from the contraction of the **Orbicularis Oris**. $e_i(p)$ returns a motion vector for moving the control point p to a point on the ellipse created when contracting the **Orbicularis Oris** [7]. Figure 6 shows the deformation of the lip due to the contraction of **Orbicularis Oris**.

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE

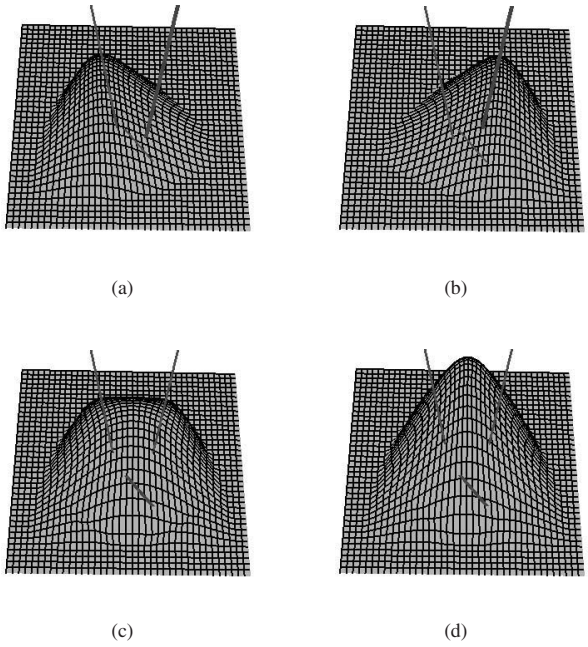


Figure 4. The effect of single muscle on the mesh (a and b); The effect of two muscles on the mesh: by simulating parallelism (c); by adding displacement (d)

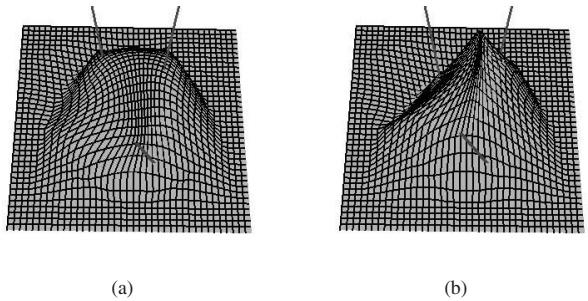


Figure 5. The effect of three muscles on the mesh: by simulating parallelism (a); by adding displacement (b)

3.3 Orbicularis Oculi

The **Orbicularis Oculi** contains two parts: the **Pars Palpebralis** that opens and closes the eyelid, and the **Pars Orbitalis** that squeeze the eye.

We have adopted the algorithm from Parke ([11]) for the closing and opening of the eyelid. The eyelid is opened and closed by combining a variation of the spherical mapping technique with linear interpolation. The squeezing of the eye is implemented with the sphincter muscle from [18].

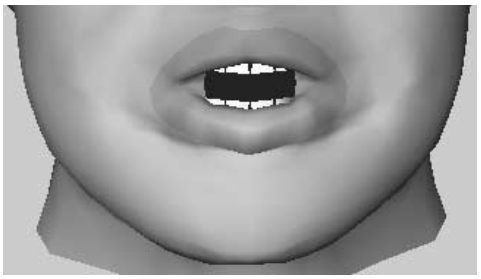


Figure 6. The deformation of the lips due to the contraction of Orbicularis Oris

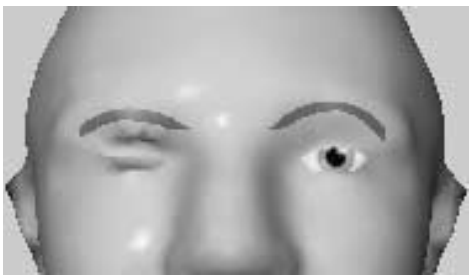


Figure 7. The closing of only one eye

Figure 8. The closing of both eyes

We note that the closing of the eyelid and the squeezing of the eye frequently occur together. The squeezing of the eye brings about the the closing of the eyelid. The closing of the eyelid also causes the squeezing of the eye when one tries to close only one of his/her eyes. We capture this relationship by modifying the contraction level of the **Pars Palpebralis** ($c_{closing}$) and the **Pars Orbitalis** ($c_{squeezing}$) in one eye as follows:

if $c_{squeezing} > 0.5c_{closing}$ then
  $c_{closing} = max(1.0, 2 \times c_{squeezing})$
else if try to close only one eye then
  $c_{squeezing} = 0.5c_{closing}$

The closing of one eye and that of both eyes are illustrated in Figures 7 and 8.

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE

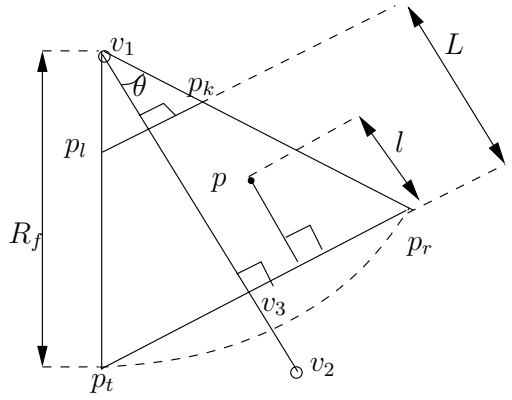


Figure 9. The zone that contains wrinkles due to the contraction of a linear muscle

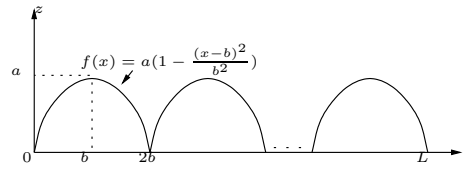


Figure 10. The wrinkle function

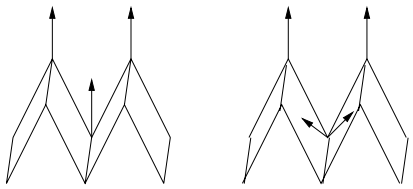


Figure 11. The "unrepresentative vertex normal" problem and its solution

3.4 Jaw rotation

The jaw is opened by rotating the vertices of the lower part of the face about a jaw pivot axis [11]. The axis of rotation is parallel to the X axis and passes through the indicated jaw pivot point. The vertices located in the lower face region are affected by jaw rotation. The lower lip, lower teeth, and the corner of the mouth rotate with the jaw.

To produce a natural oval-looking mouth, the vertices on the lower lip are rotated by different amounts. The vertices in the middle of lower lip are rotated by the same amount as the jaw rotation. The amount of rotation decreases when applied to the vertices which are closer to the corners of the mouth. The corners of the mouth are rotated by one third the jaw rotation.

The upper lip is also affected by the jaw rotation. The vertices on the upper lip are pulled down with different amounts. The amount is zero for vertices in the middle of the upper lip. This amount increases when the vertices are closer to the corners of the mouth.

4 Bulges and wrinkles

Bulges and wrinkles are created during the contraction of facial muscles. We present here a simple approach to create realistic bulges and wrinkles that works with the Waters muscle model.

For the sake of simplicity, we assume that the muscles lie parallel to the facial skin and that the heights of the wrinkles for each muscle are the same. We assign predefined values to the height of the wrinkles and the number of wrinkles (N_w) created by the contraction of each muscle. These values might be computed taking the volume preservation and a model of skull into consideration.

The wrinkle amplitude is calculated for all vertices that are originally (before applying the displacement caused by the muscle contraction) inside the $p_l p_k p_r v_3 p_t$ region (see Figure 9), where the distance from p_l and p_k to $p_t p_r$ is:

$L = \frac{3}{4}|v_1 v_3| = \frac{3}{4} R_f cos(\theta)$

The wrinkle amplitude at a vertex p is a function of the distance l from p to $p_t p_r$. We use a series of parabolas to represent this function (see Figure 10), which is described as follows:

$amp(p) = f(l) = a(1 - \frac{(u-b)^2}{b^2})$

where a is the height of the wrinkles,

$b = \frac{L}{2N_w}$

and

$u = l - \lfloor \frac{xN_w}{L} \rfloor \frac{L}{N_w}$

As $\lfloor\ \rfloor$ truncates a real number to the biggest integer number that is smaller than it, the above equation periodically maps l into $(0, 2b)$ with a frequency of N_w.

For vertices that are inside the zone of influence of multiple muscles, only the maximum wrinkle amplitude caused by these muscle is taken. The wrinkle amplitudes are applied after the vertices are displaced by the muscle contractions.

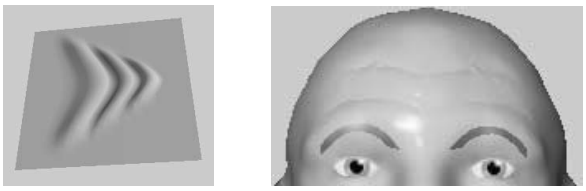
To make the wrinkles move visible, we have to prevent the "unrepresentative vertex normal" defect of interpolated shading (see Figure 11) for vertices that lie on the "internal foot" of the wrinkles. These vertices are the ones with original distance to $p_t p_r$ being:

$2b, 4b, \ldots, L - 2b$

Instead of using these vertices' average vertex normal, we use the normal of the triangular polygons that contain them.

An example of the wrinkles is shown in Figure 12(a). The wrinkles created on the forehead are shown in figure 12(b).

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE



(a) (b)

Figure 12. The wrinkles due to the muscle contraction

5 The improvements on the animation

We have improved the animation speed by improving the muscle model with a "cut-off" technique. We have also used some classic implementation techniques to improve the animation.

The linear muscle model is suitable for 3D face models with a small number of polygons. When the number of polygons increases, enabling more realistic facial expression, the muscle model's heavy computation prevents real-time animation. We introduce a "cut-off" technique to improve the animation speed.

The core of the implementation of the muscle model by Waters [19] looks like:

```
for all vertices
    if the vertex is inside the muscle's zone of influence
        then calculate and apply the displacement of the vertex
```

Let's analyze the complexity of the algorithm. For each muscle, the algorithm has to check for each vertex in the mesh if the vertex is inside the muscle's zone of influence. To do this, the algorithm has to calculate the distance from the vertex to the head of the muscle pv_1, and has to calculate the angle pv_1p_m (Figure 3). Finally, a cosine function has to be computed. The complexity of the algorithm depends on the number of vertices it has to check to see whether they are inside the zone of influence.

Our 3D face has been divided into regions (see Section 2.1), and with the knowledge of the position of the facial muscles, we know which regions a muscle will have influence on. Each muscle is associated with a flag, indicating which regions of the face it has effect on. The original implementation is now modified as follows:

```
for all vertices
    if the vertex is inside the region that the muscle has effect on
        if the vertex is inside the muscle's zone of influence
            then calculate and apply the displacement of the vertex
```

This technique first eliminates all the vertices that are outside the region that the muscle has effect on by a single check of the description file (see Section 2.1). As the number of vertices increases, this will hugely reduce the computation of checking lengths and angles. The improvement on our model of 2480 vertices is shown in Table 1. We increase the animation speed still further by applying some classic implementation techniques. First of all, the data structure plays an important role in reducing computation time. The vertices are stored separately from the polygons. The polygons contain only links to the vertices. Computation time is reduced because the calculation of the displacement of each vertex does not have to be duplicated for each of the polygon. This also prevents the recalculation of the average vertex normal of every polygon vertex for Phong shading. Second, all the condition checking with division and i-th root mathematic operators are replaced by the ones with multiplication and i-th power ones. The improvement after using all above techniques is also shown in Table 1.

	Animation speed
Before improv.	20.5fps
After muscle model improv.	30.5fps
After all improv.	35.0fps

Table 1. The result of the improvement on the animation (on a Pentium III 800Mhz, 256MB RAM, NVidia GeForce3 video card)

6 Conclusion

In this paper, we have described a simple 3D face model that can display realistic facial expressions in real-time. We have tested it to display emotional facial expressions. Emotion intensities are converted to the muscle contraction levels by the rules described in [3]. The muscles contraction levels are then applied in the face model to generate facial expressions. Some examples of emotional facial expression are shown in Figures 14, 15 and 16. From these figures, the natural jaw rotation can be seen in the surprise face. Also in the surprise face, the combination of the **Frontalis** muscles generates wrinkles in the forehead. The bulges can be seen very clearly in the happy face. Finally, the bulges and wrinkles in some regions can be seen in the sad face.

With the extension of Waters' muscle model to generate bulges and wrinkles and to combine multiple muscle actions, we have successfully created facial expressions in real-time. Maintaining a certain level of simplicity, we have achieved fast animation on a regular personal computer. Our approach is easy to apply to other facial meshes as the muscle representation is independent of the face mesh.

There are some issues that we want to improve on our model in the future. First of all, an automatic mesh reduction algorithm would reduce the requirement for a human factor when processing a new face mesh. Secondly, we

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE

want to implement an algorithm to automatically map the region division into a new face mesh. Finally, we want to test how texture mapping would affect the quality of facial expressions and the speed of animation.

References

[1] Argyle, M. (1990), Bodily Communication, London: Routledge.

[2] Basmajian, J.V. (1974), "Muscles Alive", The Williams & Wilkins Co., Baltimore.

[3] Bui,T.D., Heylen,D., Poel, M. and Nijholt, A. (2001), "Generation of facial expressions from emotion using a fuzzy rule based system", In: Lecture Notes in Artificial Intelligence 2256, M. Stumptner, D. Corbett & M. Brooks (eds.), Springer, Berlin, 83- 94.

[4] Cohen, M. M., & Massaro, D. W. (1993), "Modeling coarticulation in synthetic visual speech", In N. M. Thalmann & D. Thalmann (Eds.) Models and Techniques in Computer Animation, Tokyo: Springer-Verlag.

[5] Khler, K., J. Haber and H.-P. Seidel (2001), "Geometry-based muscle modeling for facial animation", Proceedings Graphics Interface 2001, pp. 37-46.

[6] Kalra P, Mangili A, Magnenat-Thalmann N, Thalmann D (1992), "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", Proc. Eurographics '92, Cambridge, U.K., Computer Graphics Forum, Vol. 2, No. 3, pp. 59-69.

[7] Scott A. King, Richard E. Parent, and Barbara Olsafsky (2000), "An Anatomically-Based 3D Parametric Lip Model to Support Facial Animation and Synchronized Speech", Proceedings of Deform 2000, 29-30 November, Geneva, pp. 7-19, 2000.

[8] Rolf M. Koch, Markus H. Gross, Friedrich R. Carls, Daniel F. von Bren, Yoav I. H. Parish (1996), "Simulating Facial Surgery Using Finite Element Models", In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, pp. 421-428, 1996.

[9] Rolf M. Koch, Markus H. Gross, Albert Bosshard (1998), "Emotion Editing using Finite Elements", Computer Graphics Forum 17(3): 295-302, 1998.

[10] Junyong Noh & Ulrich Neumann (1998), "A Survey of Facial Modeling and Animation Techniques", USC Technical Report 99-705.

[11] F.I. Parke (1974), "A Parameteric Model for Human Faces", PhD thesis, University of Utah, Salt Lake City, UT, December 1974. UTEC-CSc-75-047.

[12] Parke, F. I. (1991), "Techniques for facial animation", In N. Magnenat-Thalmann and D. Thalmann (Eds.), New Trends in Animation and Visualization, John Wiley, Chichester, 229-241.

[13] Parke, F. I. and Waters, K. (1994), Computer Facial Animation, AK Peters. ISBN 1-56881-014-8.

[14] S. Pasquariello & C. Pelachaud (2001), "Greta: A Simple Facial Animation Engine", 6th Online World Conference on Soft Computing in Industrial Appications, Session on Soft Computing for Intelligent 3D Agents, Septembre 2001.

[15] S. Platt & N. Badler, "Animating facial expression", Computer Graphics, 1981, vol. 15(3) pp. 245-252.

[16] Terzopoulos, D. and Waters, K., "Physically-Based Facial Modeling, Analysis, and Animation", The Journal of Visualization and Computer Animation, Vol.1, pages 73-80, December 1990.

[17] Carol L. Wang (1993), "Langwidere: A Hierarchical Spline Based Facial Animation System with Simulated Muscles", Master's thesis, University of Calgary, October 1993.

[18] Waters, K. (1987), "A muscle model for animating three-dimensional facial expressions", Computer Graphics (SIGGRAPH'87), 21(4), July, 17-24.

[19] http://crl.research.compaq.com/publications/books/ waters/Appendix1/appendix1.html

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE

Figure 13. The neutral face

Figure 14. The face model displays surprise

Figure 15. The face model displays happiness

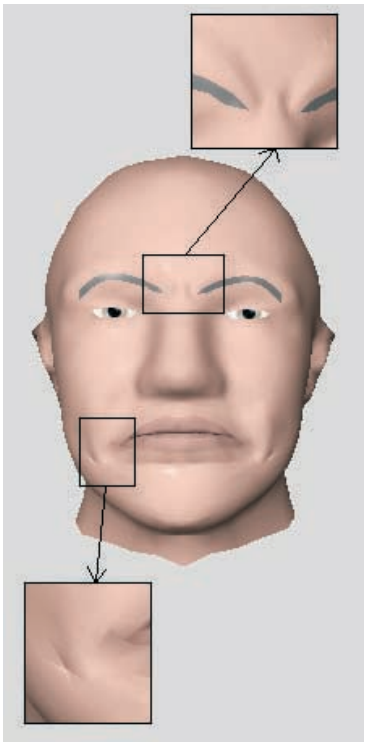


Figure 16. The face model displays sadness

Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)
1087-4844/03 $17.00 © 2003 IEEE