

# Emotion Editing using Finite Elements

Rolf M. Koch, Markus H. Gross, Albert A. Bosshard

Computer Science Department  
ETH Zurich, Switzerland  
e-mail: {koch, grossm}@inf.ethz.ch  
<http://www.inf.ethz.ch/department/IS/cg>

---

## ABSTRACT

*This paper describes the prototype of a facial expression editor. In contrast to existing systems the presented editor takes advantage of both medical data for the simulation and the consideration of facial anatomy during the definition of muscle groups. The  $C^1$ -continuous geometry and the high degree of abstraction for the expression editing sets this system apart from others. Using finite elements we achieve a better precision in comparison to particle systems. Furthermore, a precomputing of facial action units enables us to compose facial expressions by a superposition of facial action geometries in real-time. The presented model is based on a generic facial model using a thin plate and membrane approach for the surface and elastic springs for facial tissue modeling. It has been used successfully for performing facial surgery simulation. We illustrate features of our system with examples from the Visible Human Dataset.<sup>TM</sup>*

**Additional Keywords and Phrases:** Facial Modeling, Emotion Editing.

---

## 1. Introduction

Because of its variability the human face is a very important means of interpersonal communication [6]. Due to the progress in computer animation researchers have developed facial models since the 70's in order to perform facial animation, for a survey see [16]. During the following two decades these models have been refined to handle arbitrary facial meshes, mostly obtained from 3D laser range scanners [12,15], in an anatomically accurate and realistic way.

In the context of craniofacial surgery simulation face models have been developed using the Finite Element approaches of various types [11,10,17]. Especially in [11] a skull and face representation is included and the volume behaviour is performed by a 3D spring mesh, whose spring elasticity is computed from a segmentation of the facial tissue based on CT-volume data. Changes in the facial surface, that result from manipulations of the underlying skull are simulated under control of the finite element method. The higher order  $C^1$ -continuous facial surface representation is preferred for the following reasons:

- Convergence of the solution for coarse finite element discretizations.

- Conformity of the finite elements with the differential equations that have to be solved.
- Smooth polynomial surface for subsequent high quality rendering schemes.

Besides the modeling of facial changes generated by surgical procedures on the skull it is desirable to synthesize emotions on the post-surgical face in order to get an impression of facial actions after surgery. Therefore, we developed the emotion editor presented in this paper. It is based on our generic facial model and the prior work of Waters et al. [20,1].

For reasons of data availability our prototype is tested on the well known Visible Human Data Set [14] but is applicable on arbitrary faces. Here, volume data can be used for skull extraction and tissue stiffness computation. The following section expands on the different procedures, data processing steps and the general set-up of our facial emotion editing system shown in figure 1. Section 3 explains some basics concerning the muscle model employed and the facial action coding system as well as the FEM pipeline proposed. Section 4 sheds light on several preprocessing steps for the generation of the physically-based facial

model and section 5 addresses the precomputing of facial action units. A detailed description of the emotion editor follows and some results conclude the paper.

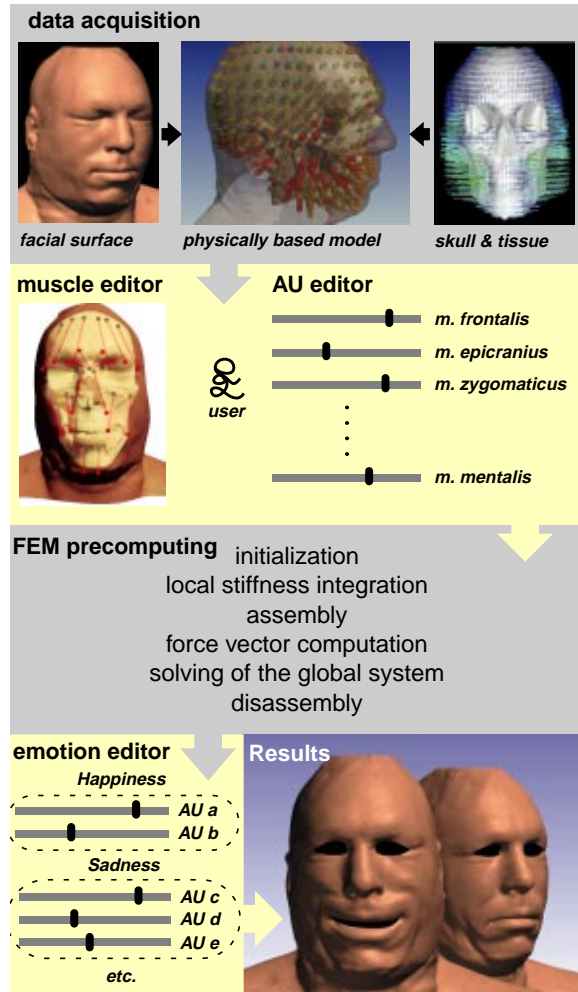


Figure 1: System overview.

## 2. System Overview

The animation prototype, presented in figure 1, is divided into data acquisition, preediting, the FEM engine, and the emotion editor.

Using the original data set consisting of surface scan and volume data, the data acquisition builds up a physically-based model of the human face combining a higher order polynomial FEM surface with a mesh of springs attached to the skull. If a CT-scan is available, the skull geometry and the tissue stiffness will be derived from a tissue segmentation based on the volume data. Otherwise, a default skull non-proportionally scaled to fit into the face and tissue stiffness values obtained from a template model can be applied.

The muscle editing process allows the definition of origins and attachment points of the facial muscles and the user combines a weighted sum of muscle contractions into one action unit (AU) during the AU editing.

The FEM engine precomputes the facial appearance of each action unit in order to obtain displacement fields for the individual face. It comprises computations for local and global stiffness matrices, assembling, solving and disassembling to obtain the new facial shape. In the emotion editor the user combines several action units to synthesize an individual expression. This is accomplished interactively and in real-time.

Further interaction is required to define boundary conditions and the stretching and bending tensors crucial to achieve robust results. Finally the facial surface is subsampled and rendered under the guidance of Alias.™

## 3. Basics

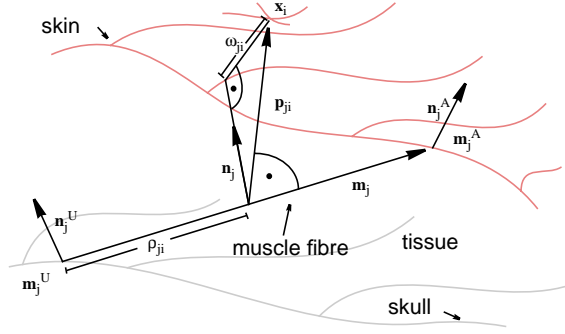
### 3.1. The Muscle Model

The muscular system of the human face is comprised of approximately 50 muscles. They are mostly attached with both the skull and the facial tissue. A special case are sphincter muscles around the eyes and mouth which are only grown together with the tissue.

Each mimic muscle consists of approximately 50 fibres, that increase its influence area on the facial surface. During the editing procedure the user defines only the central fibres. The following section addresses the generation of the entire muscular system from the central muscle fibres and from the facial and skull geometry. Here, we follow the approaches from Lee et al. [13].

On contraction facial regions close to the skin attachment point of a muscle are affected. *The muscular influence decreases with both the distance perpendicular to the muscle vector  $\rho_{ji}$  and with the distance from the muscle attachment point in direction to the muscle origin  $\omega_{ji}$ .* The following paragraphs describe the computation of the values  $\rho_{ji}$  and  $\omega_{ji}$  for an arbitrary point on the facial surface. Using the following definitions

- $\mathbf{x}_i$ : arbitrary surface point  $i$  at the face
- $\mathbf{m}_j^A$ : attachment of muscle  $j$  at the facial surface
- $\mathbf{m}_j^U$ : origin of muscle  $j$  at the skull
- $\mathbf{m}_j$ : muscle vector  $j$
- $\mathbf{n}_j^A$ : surface normal at  $\mathbf{m}_j^A$
- $\mathbf{n}_j^U$ : surface normal at  $\mathbf{m}_j^U$
- $\mathbf{n}_j$ : muscular (averaged) normal of muscle  $j$
- $\mathbf{p}_{ji}$ : distance between  $\mathbf{m}_j$  and  $\mathbf{x}_i$



**Figure 2:** Variables defining the influence of a facial muscle at an arbitrary facial surface point  $x_i$ .

$\rho_{ji}$  and  $\omega_{ji}$  are calculated as

$$\mathbf{m}_j = \frac{\begin{pmatrix} U \\ \mathbf{m}_j - \mathbf{m}_j^A \end{pmatrix}}{\left\| \begin{pmatrix} U \\ \mathbf{m}_j - \mathbf{m}_j^A \end{pmatrix} \right\|}, \quad \mathbf{n}_j = \frac{\begin{pmatrix} U \\ \mathbf{n}_j - \mathbf{n}_j^A \end{pmatrix}}{\left\| \begin{pmatrix} U \\ \mathbf{n}_j - \mathbf{n}_j^A \end{pmatrix} \right\|} \quad (1)$$

$$\rho_{ji} = \begin{pmatrix} U \\ \mathbf{m}_j - \mathbf{x}_i \end{pmatrix} \cdot \mathbf{m}_j \quad (2)$$

and

$$\mathbf{p}_{ji} = \mathbf{x}_i - \mathbf{m}_j^U - \rho_{ji} \mathbf{m}_j \quad (3)$$

$$\omega_{ji} = \left\| \mathbf{p}_{ji} - (\mathbf{p}_{ji} \cdot \mathbf{n}_j) \cdot \mathbf{n}_j \right\|. \quad (4)$$

$\rho_{ji}$  and  $\omega_{ji}$  weight the influence of muscle  $j$  at vertex  $i$  separately for length distance  $\varepsilon_{ji}$  and width distance  $\gamma_{ji}$ :

$$\varepsilon_{ji} = \frac{\rho_{ji}}{\left\| \begin{pmatrix} A \\ \mathbf{m}_j^A - \mathbf{m}_j^U \end{pmatrix} \right\|} \quad (5)$$

$$\gamma_{ji} = \begin{cases} \frac{\omega_{ji}}{\omega_j} & \text{if } \omega_{ji} < \omega_j \\ 0 & \text{else} \end{cases} \quad (6)$$

$\varepsilon_{ji}$  defines the longitude distance of vertex  $i$  to muscle  $j$  normalized to values between 0 and 1. A value around 0, or 1 signifies that vertex  $i$  lies close to the origin, or close to the attachment point, respectively. The influence of the muscle  $j$  increases with  $\varepsilon_{ji}$ . The constant  $\kappa_j$  defines the strength of muscle  $j$ . A decreasing value of  $\kappa_j$  increases the influence along the longitude.

$\gamma_{ji}$  is defined in  $[0, 1]$  and represents the latitude distance between vertex  $i$  and muscle  $j$ .  $\omega_j$  defines the width of muscle  $j$ . Increasing  $\gamma_{ji}$  results in decreasing the influence of muscle  $j$ . Figure 4 shows the weighting functions  $\Theta_1$  and  $\Theta_2$

$$\Theta_1(\varepsilon_{ji}) = \sin\left(\frac{\pi}{2} \varepsilon_{ji}^{\kappa_j}\right) + \frac{1}{8} \left( \sin\left(2\pi \varepsilon_{ji}^{\kappa_j} - \frac{\pi}{2}\right) + 1 \right) \quad (7)$$

$$\Theta_2(\gamma_{ji}) = \frac{1}{2} \left( \cos\left(\pi \gamma_{ji}^{1.8}\right) + 1 \right) \quad (8)$$

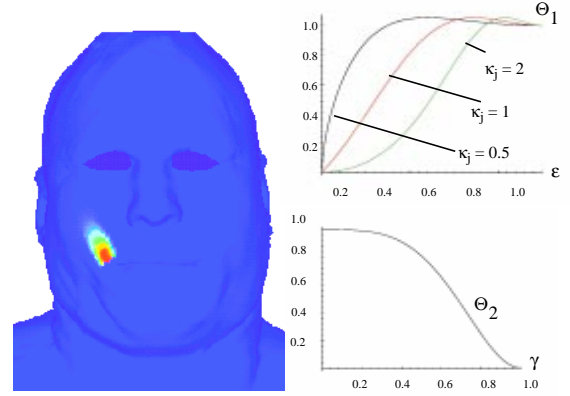
together with the area of influence of the muscle *zygomaticus major*.

The muscular force vector applied at vertex  $i$  can be computed as

$$\mathbf{f}_{ji} = \Theta_1(\varepsilon_{ji}) \Theta_2(\gamma_{ji}) \mathbf{m}_j. \quad (9)$$

Assuming that  $N$  muscle fibres are attached at the surface node  $i$  the resulting force  $\mathbf{f}_i$  at vertex  $i$  has to be computed by superposing  $\mathbf{f}_{ji}$  of each muscle fibre  $j$  as

$$\mathbf{f}_i = \sum_{j=1}^N \mathbf{f}_{ji}. \quad (10)$$



**Figure 4:** Left: Area of influence (*zygomaticus major*), right:  $\Theta_1$  for different values of  $\kappa_j$  and  $\Theta_2$ .

### 3.2. The “Facial Action Coding System (FACS)”

Besides a physically-based facial model the simulation of facial emotions requires a mapping of the desired facial expression into muscle activations. For this purpose the “Facial Action Coding System” (FACS) of Ekman and Friesen [5] has become established. Based on photographs of facial expressions they investigated in fundamental psychological tests the relationship between facial expressions and emotions. The results of this research is a unique categorization of the emotions (*surprise, fear, disgust, anger, happiness, sadness*) into facial base expressions.

An individual facial play is defined by the activation of properly chosen action units of the FACS which comprises about 66 AUs categorized as follows:

- I. AUs of the upper face
- II. AUs of the lower face
  - a) Horizontal Action Units (e.g. dimpler)
  - b) Vertical Action Units
  - c) Diagonal Action Units (e.g. lip corner puller)
  - d) Orbital Action Units (e.g. lip pressor)

III. Mixed AUs (e.g. sticking out the tongue)

IV. Head and eye position

The groups I and II are playing a leading role in the simulation of facial emotions. Table I contains the action units chosen for the simulation and the designation of the corresponding muscles.

TABLE I  
Action units needed, from [5]

AU	Name in FACS	active muscle groups
1	inner brow raiser	<i>frontalis, pars medialis</i>
2	outer brow raiser	<i>frontalis, pars lateralis</i>
3	brow lowerer	<i>depressor glabellae; depressor supercillii; corrugator</i>
4	brow lowerer	<i>depressor glabellae; depressor supercillii; corrugator</i>
5	upper lid raiser	<i>levator palpebrae superioris</i>
6	cheek raiser	<i>orbicularis oculi, pars orbitalis</i>
7	lid tightener	<i>orbicularis oculi, pars palpebralis</i>
8	lips toward each other	<i>Orbicularis oris</i>
9	nose wrinkler	<i>levator labii superioris, alaeque nasi</i>
10	upper lid raiser	<i>levator labii superioris, caput infraorbitalis</i>
11	nasolabial furrow deepener	<i>zygomaticus minor</i>
12	lip corner puller	<i>zygomaticus major</i>
13	cheek puffer	<i>caninus</i>
14	dimpler	<i>buccinator</i>
15	lip corner depressor	<i>triangularis</i>
16	lower lip depressor	<i>depressor labii inferioris</i>
17	chin raiser	<i>mentalis</i>
18	lip puckerer	<i>incisivii labii superioris; incisivus labii inferioris</i>
19	lip stretcher	<i>risorius</i>
20	lip funneler	<i>orbicularis oris</i>
22	lip tightener	<i>orbicularis oris</i>
23	lip pressor	<i>orbicularis oris</i>
24	lips part	<i>depressor labii, or relaxation of the mentalis or orbicularis oris</i>
25	jaw drops	<i>masseter</i>
26	mouth stretches	<i>pterygoids; digastric</i>
27	lips suck	<i>orbicularis oris</i>
28	nostril dilator	<i>nasalis, pars alaris</i>
38	nostril compressor	<i>nasalis, pars transversa and depressor septi alaeque nasi</i>
39	lids drop	<i>relaxation of the levator palpebrae superioris</i>
41	eyes slit	<i>orbicularis oculi</i>
42	eyes close	<i>relaxation of the levator palpebrae superioris</i>
43	squint	<i>orbicularis oculi, pars palpebralis</i>
44	blink	<i>relaxation of the levator palpebrae and Contraction des Orbicularis oculi, pars palpebralis</i>
45	wink	<i>orbicularis oculi</i>

### 3.3. The FEM Engine

The FEM engine performs the central computation unit of the presented emotion editor. Its implementation is already described in details in [11]. The goal is the computation of a solution function  $\mathbf{w}(u,v)$  defined in the  $u,v$  surface parameter space  $(u, v) \in \Omega$ :

$$\mathbf{w}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}. \quad (11)$$

The potential energy  $\Pi_{Sf}$  of  $\mathbf{w}(u,v)$  defined as

$$\Pi_{Sf} = \int_{\Omega} \left[ \alpha_{11} \mathbf{w}_u^2 + 2\alpha_{12} \mathbf{w}_u \mathbf{w}_v + \alpha_{22} \mathbf{w}_v^2 + \beta_{11} \mathbf{w}_{uu}^2 + 2\beta_{12} \mathbf{w}_{uv}^2 + \beta_{22} \mathbf{w}_{vv}^2 \right] - 2\mathbf{f} \cdot \mathbf{w} \, d\Omega \quad (12)$$

with

$$\begin{aligned} \mathbf{w}_u &: \frac{\partial \mathbf{w}}{\partial u} \\ \mathbf{w}_{vv} &: \frac{\partial^2 \mathbf{w}}{\partial v^2} \end{aligned} \quad (13)$$

and

$$\begin{aligned} \alpha &= \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{12} & \alpha_{22} \end{bmatrix} \quad (\text{stretching tensor}) \\ \beta &= \begin{bmatrix} \beta_{11} & & \\ & \beta_{22} & \\ & & \beta_{12} \end{bmatrix} \quad (\text{bending tensor}) \end{aligned} \quad (14)$$

has to be minimized under constraint of the essential boundary conditions  $\mathbf{w}_{bc}$  on the surface region  $bc \subset \Omega$ , i.e.  $\mathbf{w}$  is supported on  $bc$  with

$$\mathbf{w}(bc) = \mathbf{w}_{bc}. \quad (15)$$

The natural boundary conditions encompass the forces  $\mathbf{f}$  applied to the surface. Using the variational approach

$$\delta \Pi_{Sf} = 0 \quad (16)$$

the Euler-Lagrange differential equations of  $\mathbf{w}$  can be derived as

$$\begin{pmatrix} (\beta_{11} \mathbf{w}_{uu})_{uu} + (\beta_{12} \mathbf{w}_{uv})_{uv} + (\beta_{22} \mathbf{w}_{vv})_{vv} \\ -(\alpha_{11} \mathbf{w}_u + \alpha_{12} \mathbf{w}_v)_u - (\alpha_{12} \mathbf{w}_v + \alpha_{22} \mathbf{w}_v)_v \end{pmatrix} = \mathbf{f} \quad (17)$$

to describe the equilibrium condition between the applied forces and the internal surface energy.

In order to solve the presented minimization problem the method of the finite elements computes an approximation of the desired function in the following steps:

- *Discretization* of the surface into discrete primitives. In our case this is performed inherently by scanning of the facial surface and subsequent mesh reduction.
- *Creation* of a vector of basis functions (interpolation functions)  $\Phi^{(m)}$  whose linear combination performs a piecewise approximation of the desired surface function. For each patch  $m$  the surface function  $\mathbf{w}^{(m)}$  can be approximated as  $\mathbf{w}^{(m)} \approx \hat{\mathbf{w}}^{(m)} = \Phi^{T(m)}(\mathbf{a}_{init} + \mathbf{a}^{(m)})$  where  $\mathbf{a}_{init}^{(m)}$  denotes the initial node vector of the original facial surface and  $\mathbf{a}^{(m)}$  a displacement vector containing the changes of the facial surface caused by muscle forces. Triangular base functions well suited for this purpose are the Hermite type  $\mathbf{N}^{12}$  presented in [11, 21].
- *Computation* of the local stiffness matrix  $\mathbf{K}^{(m)}$ . The surface energy for each patch  $m$  can then be obtained as  $\Pi_{sf}^{(m)} = \mathbf{a}^{(m)T} \mathbf{K}^{(m)} \mathbf{a}^{(m)} - \mathbf{F}^{(m)T} \mathbf{a}^{(m)}$  with the external energy  $\mathbf{F}^{(m)} \approx 2 \int_{\Omega^{(m)}} \Phi^{T(m)} \mathbf{f}^{(m)} d\Omega^{(m)}$ . Since  $\mathbf{a}_{init}^{(m)}$  defines the state of rest it is not taken into consideration for further computations.
- *Assembly* of the patch stiffness matrices  $\mathbf{K}^{(m)}$  and the local loads  $\mathbf{F}^{(m)}$  into a global stiffness matrix  $\mathbf{K}$  and a global load vector  $\mathbf{F}$ .
- *Solving* of the global equation system  $\mathbf{K}\mathbf{a} = \mathbf{F}$ . Since  $\mathbf{K}$  is symmetric and positive definite the conjugate gradient approach [9] together with an incomplete Cholesky preconditioning is most appropriate for this purpose.
- *Disassemble* for each patch  $m$  the global node vector  $\mathbf{a}$  into  $\mathbf{a}^{(m)}$ .
- *Visualization* of the resulting patch surfaces  $\hat{\mathbf{w}}^{(m)} = \Phi^{T(m)}(\mathbf{a}_{init}^{(m)} + \mathbf{a}^{(m)})$  by a sampling of the shape functions and subsequent rendering.

Further detailed information concerning the finite element method can be found in [11, 21, 2].

#### 4. Data Acquisition and Preprocessing

##### 4.1. Geometry Extraction and Reduction

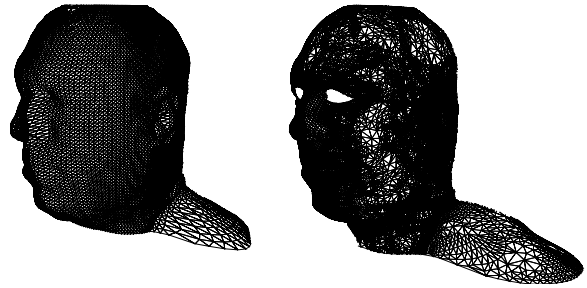
As stated earlier we use the VHD as a reference model. For the presented simulations the high resolution cryosections and computer tomographies of a lower resolution are employed. The cryosections are used for the extraction of the facial surface - range scans are not available for the VH - and the computer tomographies hold for the evaluation of the skull surface and the tissue segmentation.

The facial surface is reconstructed from the volume data set as follows:

- Scaling, clipping and registering of the cryosections and the computer tomographies with respect to reference points.
- Edge detection performed by the Canny operator [3] on the blue channel of each cryosection.
- Simulating a range scan for the computation of the facial surface. Therefore we intersect the Canny edges with rays through the center in a given angular resolution. The respective range value is found by computing the Euclidean distance between the center of projection and the intersection points.
- The cryosection number, the angle of the viewing ray and the computed range define the discrete facial geometry in a cylinder coordinate system.
- For further computations the high resolution facial geometry is produced by a triangulation.

The global parametrization produced inherently is used for the construction of  $C^1$ -continuous interpolation functions for the FEM engine.

For performance reasons a mesh reduction [19] is crucial. We employ a slightly modified version of Schroeders algorithm [18]. Figure 5 shows two facial surface meshes containing approximately the same number of vertices for both a regular and an adaptively reduced geometry.



**Figure 5:** Adaptive triangular meshes of the VHD. (Data source: Visible Human Project, Courtesy National Library of Medicine). Left: Regular mesh (20000 vertices), right: Adaptively reduced mesh (19680 vertices).

Although both meshes are of the same complexity the latter models fine grain surface details more accurately.

The skull mesh is extracted from the CT similarly to the facial surface from the cryosections. If there is no volume data available a template skull - scaled and registered to the face - must be employed.

##### 4.2. Facial Elasticity

In order to account for individual tissue behaviour which is varying with age, gender, and ethnic group, it is crucial to distinguish between different facial tissue types [7,8]. Therefore the physically-based facial model presented in [11] comprises a spring mesh to simulate soft tissue behaviour. The spring constants are extracted from a tissue segmentation and the surface elasticity is set interactively. We base the computation of the spring stiffness on a segmentation of the underlying volume data. In the general case of

CT data, much work has been done to provide appropriate tissue segmentations, but intensity-based methods still do a good job. For further details, we refer to [11].

1) Computation of the spring stiffness

The spring mesh is constructed by projecting the facial surface triangulation onto the skull and connecting both surfaces with main springs  $s_{i,i}$  and strut springs  $s_{i,j}$ . In order to compute the spring stiffness we have to consider that different tissue types are penetrated between the facial surface attachment  $\mathbf{x}_s^{(i)}$  and the origin  $\mathbf{x}_b^{(j)}$  on the skull. We cope with this problem by decomposing the spring into a set of smaller springs connected in series. Let  $c(\mathbf{x})$  be the tissue elasticity function and  $\mathbf{x}$  the spatial parameter. Then the stiffness  $c_{i,j}$  can be computed analytically as

$$\frac{1}{c_{i,j}} = \int_{\mathbf{x}_s^{(i)}}^{\mathbf{x}_b^{(j)}} \frac{1}{c(\mathbf{x})} d\mathbf{x} \quad (18)$$

With a discretization into  $L$  springs of individual lengths  $l_1, \dots, l_L$  and stiffnesses  $c_1, \dots, c_L$  the upper integral collapses for a segmented volume data set into

$$\frac{1}{c_{i,j}} = \frac{1}{l} \sum_{h=1}^L \frac{l_h}{c_h} \text{ with the spring length } l = \sum_{h=1}^L l_h \quad (19)$$

Elasticity coefficients for several tissue types are given in [7,8].

2) Definition of the Surface Tensors

Similar to the spring mesh the facial surface resists to deformation. In our membrane plate surface model this behaviour can be manipulated interactively by painting stretching and bending parameters  $\alpha$  and  $\beta$  on the facial surface. Commercially available software packages, for instance Studio Paint 3D can be used for the process of surface tensor definition.

4.3. Boundary Conditions

Besides the surface tensors  $\alpha$  and  $\beta$  the rigid and non-rigid parts of the facial surface has to be defined. The definition of boundary conditions is crucial for the following reasons:

- Solubility of the governing equations.
- Suppression of the deformation of non-rigid facial regions, e.g. the bridge of the nose.
- Decrease of the overall degree of freedom.

Figure 6 shows that the resulting FEM surface varies as a function of the boundary conditions set for a constant force vector. As depicted in figure 7 and similar to the setting of the surface tensors the user defines the boundary conditions via surface texture mapping.

4.4. Interactive Muscle Definition

The main preprocessing step of anatomy-based facial animation is the definition of the central muscle fibres. Since this task cannot be fully automated, user interaction is necessary. Figure 8 presents the facial skin, the skull of the

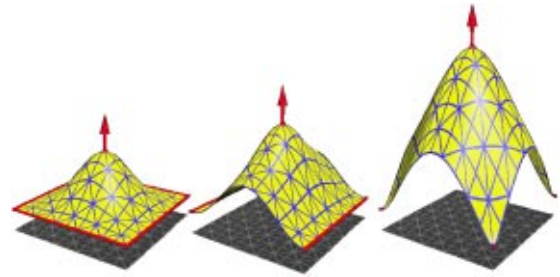


Figure 6: Variation of the surface depending on the boundary conditions.



Figure 7: Setting of the boundary conditions by defining a texture map. Left: Texture map, right: Visualized boundary conditions.

VHD, and the central muscle fibres defined by the user. For reasons of clarity the sphincter muscles are not depicted. In order to insert new muscles into the model the facial and skull surfaces have to be represented in Alias. A geometry object comprising all muscle fibres is scaled to the size of the face and registered manually. By picking the ends of each fibre and the gravity option turned on, the end of each muscles can easily be moved to a surface vertex.

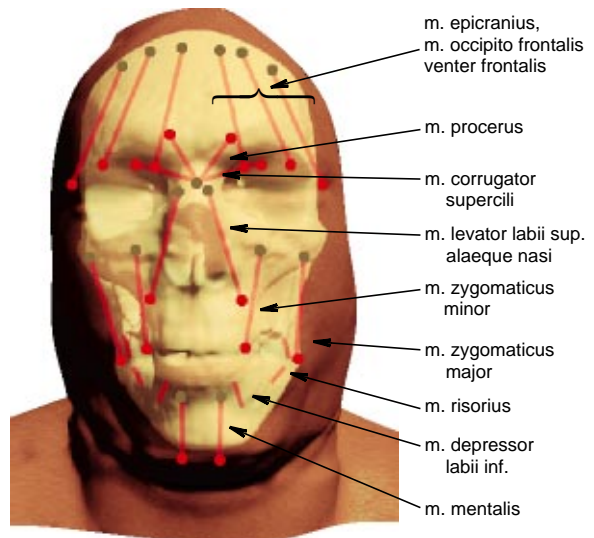


Figure 8: Interactive muscle definition and attachment.

## 5. Precomputing of the Facial Actions

The expensive FEM computation would exclude the system from real-time application in facial animation. Therefore, many researchers are focussed on the development of real-time linear elastic elements [4]. Without any loss of accuracy the editing process of a facial expression can be accelerated by using precomputed displacement fields for each facial action or at least for each base emotion, such as happiness, sadness, surprise, anger, fear, or disgust.

As stated earlier, the resulting facial shape is determined by a displacement vector  $\mathbf{a}$  which weights the interpolation functions of each surface patch and represents the difference between the original and the resulting face. Recall that the right hand side of the linear equation system  $\mathbf{K}\mathbf{a} = \mathbf{F}$  represents the surface loads caused by muscle contractions. Assuming linear settings and let  $\mathbf{F}_{AU_i}$  and  $\mathbf{F}_{AU_j}$  be the surface loads resulting from different facial action units  $AU_i$  and  $AU_j$ , respectively, we use the precomputed node vectors  $\mathbf{a}_{AU_i}$  and  $\mathbf{a}_{AU_j}$  rather than recomputing the facial shape for the superposition of both surface loads:

$$\mathbf{K}\mathbf{a}_{AU_i} = \mathbf{F}_{AU_i} \quad \text{and} \quad \mathbf{K}\mathbf{a}_{AU_j} = \mathbf{F}_{AU_j} \quad (20)$$

then

$$\mathbf{K}\mathbf{a} = \mathbf{F}_{AU_i} + \mathbf{F}_{AU_j} \quad \text{with} \quad \mathbf{a} = \mathbf{a}_{AU_i} + \mathbf{a}_{AU_j} \quad (21)$$

Note that the upper approach is only justified for linear elasticity. In nonlinear settings such as arising with large deformations, more sophisticated approximations have to be found [2].

For instance, each patch  $m$  of a facial expression comprising the action units  $AU_i$  and  $AU_j$  to 70% and 30% is computed as

$$\hat{\mathbf{w}}^{(m)} = \Phi^{(m)} \bullet (0.7\mathbf{a}_{AU_i}^{(m)} + 0.3\mathbf{a}_{AU_j}^{(m)} + \mathbf{a}_{init}^{(m)}). \quad (22)$$

Since the values of the polynomial interpolation functions  $\Phi$  have to be computed, the evaluation of the upper scalar product is quite expensive. In order to reduce computational costs during the emotion editing, we propose to precompute the vertex displacements of the sampled surface for each action unit. Let  $\mathbf{u}_{init}$  be the displacements of the geometry nodes of the subsampled original face and  $\mathbf{u}_{AU_k}$  the displacement for each node after the activation of an action unit  $k$ , then the resulting facial surface can be computed as

$$\mathbf{u} = 0.7\mathbf{u}_{AU_i} + 0.3\mathbf{u}_{AU_j} + \mathbf{u}_{init}. \quad (23)$$

For each vertex being translated due to the facial action 6 multiplications and 6 additions are necessary.

## 6. The Emotion Editor

The editing process comprises the definition of facial action units in a first step and the composition of the facial expression in a second step after precomputing. Figure 9 depicts the user interface of our system.

In order to define a facial action unit the user sets the activation of each muscle as well as a jaw rotation in an interactive process. This step is supported by a force vector preview and the activation area of the selected muscles.

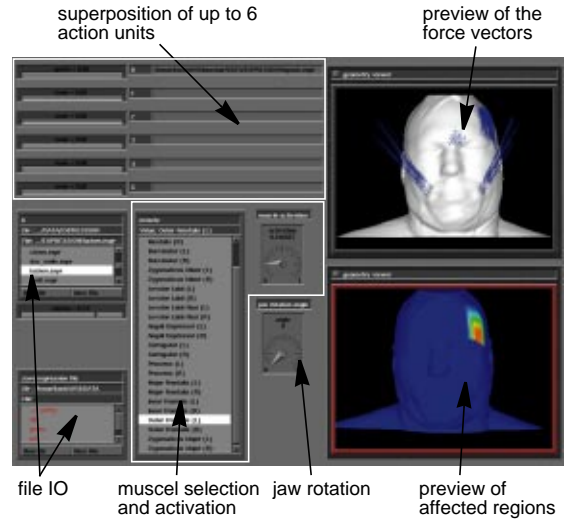


Figure 9: Components of the prototype in AVS 5.

Optionally, a jaw rotation can be added. Since the facial action coding system already defines the activation of each muscle for each action unit this task can be fully automated.

If a higher level of abstraction is needed, it is sufficient to define the muscle activation for the 6 base emotions surprise, fear, disgust, anger, happiness and sadness.

In contrast to the facial expressions caused by muscle activation the jaw rotation is obtained by manipulating the underlying skull geometry. This procedure is comparable to the simulation of craniofacial and maxillofacial surgeries presented in [11]. The skull manipulation is performed as follows: In a first step the user has to select all geometry vertices that belong to the jaw area of the skull. Then the rotation axis of the jaw bone has to be defined. Figure 10 shows the selected jaw of the VHD skull and a jaw rotation by 15 degrees. In Figure 11 the force vectors and the resulting facial surface is presented.

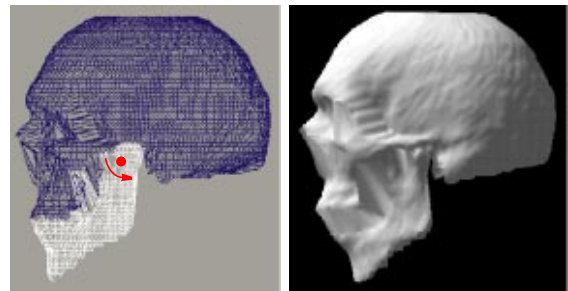
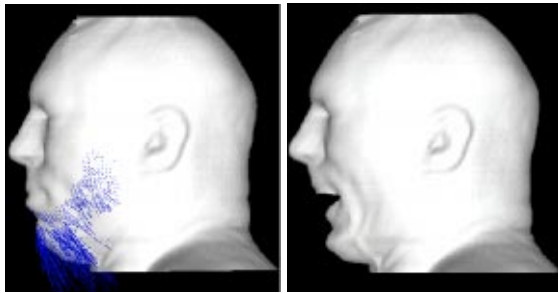


Figure 10: Left: Jaw area selected in Alias. Right: Resulting skull after jaw rotation of 15°.



**Figure 11:** *Left: Forces resulting from a jaw rotation. Right: Resulting facial surface.*

## 7. Results

Figures 12 to 15 show results obtained with the presented prototype. The editor and the FEM pipeline is programmed in AVS and the visualization is performed under Alias.

Figure 12 shows a little smile, which is created by an activation of the muscles *cygomaticus major* and *depressor labii*. In 13 the *cygomaticus minor*, *levator labii* and *levator labii nasi* are activated. Figure 14 illustrates the *anguli depressor* pulling the mouth corners downwards and 15 shows the result of a jaw rotation combined with activated *depressor labii*.



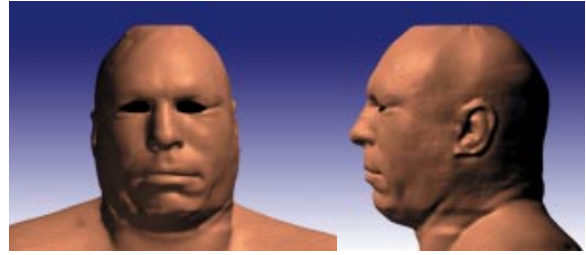
**[Koch et al.] Figure 12:** *Smile, open mouth.*



**[Koch et al.] Figure 13:** *Smile, closed mouth.*

## 8. Discussion and Future Directions

We present an emotion editor which enables us to predict deformations of the facial shape caused by emotions. Our system uses commercial tools for interactive geometric modeling and rendering. In contrast to existing approaches we use higher order polynomial finite elements that enables



**[Koch et al.] Figure 14:** *Sadness.*



**[Koch et al.] Figure 15:** *Jaw rotation only.*

us to predict a highly accurate  $C^1$ -continuous facial surface. Following Ekman and Friesen [5] we assume that each facial expression is a superposition of several AUs or base emotions whose precomputed displacement fields are used to perform facial animation. Optionally, CT, or in the case of VHD, cryosections are applied to extract tissue stiffness and skull geometry. If not available skull and tissue stiffness information can be approximated from the VHD template. A high degree of abstraction allows the facial editing to be done by the superposition of base emotions and action units.

Accounting for real volume behaviour we are currently implementing the linear FEM mixed formulation which features volume preservation.

## 9. References

- [1] F. Barke and K. Waters. *Computer Facial Animation*. AK Peters, 1996.
- [2] K.-J. Bathe. *Finite element procedures*. Prentice Hall, Englewood Cliffs, 2nd edition, 1996. Rev. of: Finite element procedures in engineering analysis. 1982.
- [3] J. F. Canny. "A computational approach to edge detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–697, 1986.
- [4] S. Cotin, H. Delingette, and N. Ayache. "Real time volumetric deformable models for surgery simulation." In K. Hohne and R. Kikinis, editors, *Visualization in Biomedical Computing (VBC '96)*, volume 1131 of *Lecture Notes in Computer Science*, pages 535–540. Springer, July 1996.
- [5] P. Ekman and W. V. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Palo Alto, 1977.
- [6] L. G. Farkas. *Anthropometry of the Head and Face*. Raven Press, 2nd edition, 1994.
- [7] Y. C. Fung. *Biomechanics: motion, flow, stress, and growth*. Springer, 1990.
- [8] Y. C. Fung. *Biomechanics: mechanical properties of living tissues*. Springer, 2nd edition, 1993.



- [9] G. Golub and C. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.
- [10] E. Keeve, S. Girod, and B. Girod. "Computer-aided craniofacial surgery." In *Proc. Computer Assisted Radiology CAR'96*, pages 757–762, June 1996.
- [11] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser, and Y. I. H. Parish. "Simulating facial surgery using finite element models." In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 421–428. ACM SIGGRAPH, Addison Wesley, Aug. 1996. held in New Orleans, Louisiana, 4-9 August 1996.
- [12] Y. Lee, D. Terzopoulos, and K. Waters. "Constructing physics-based facial models of individuals." In *Proceedings of Graphics Interface '93*, pages 1–8, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.
- [13] Y. Lee, D. Terzopoulos, and K. Waters. "Realistic face modeling for animation." In R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 55–62. ACM SIGGRAPH, Addison Wesley, Aug. 1995. held in Los Angeles, California, 06-11 August 1995.
- [14] National Library of Medicine. *The Visible Human Project*. [http://www.nlm.nih.gov/extramural\\_research.dir/visible\\_human.html](http://www.nlm.nih.gov/extramural_research.dir/visible_human.html), 1995.
- [15] F. Parke and K. Waters. *Computer Facial Animation*. A K Peters, 1993. ISBN 1-56881-014-8.
- [16] F. I. Parke. "Parameterized models for facial animation." *IEEE Computer Graphics and Applications*, 2:61–68, Nov. 1982.
- [17] S. H. M. Roth, M. H. Gross, S. Turello, and F. R. Carls. "A bernstein-bézier based approach to soft tissue simulation." In *Proc. EUROGRAPHICS'98*, 1998.
- [18] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. "Decimation of triangle meshes." In *Proceedings of SIGGRAPH '92*, pages 65–70. ACM SIGGRAPH, 1992.
- [19] O. G. Staadt, M. Gross, and R. Weber. "Multiresolution compression and reconstruction." In *Proceedings of IEEE Visualization 1997*, pages 337–346. IEEE, 1997.
- [20] K. Waters. "A muscle model for animating three-dimensional facial expression." In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 17–24, July 1987.
- [21] O. C. Zienkiewicz. *The Finite Element Method*, volume 1–2. McGraw-Hill, 4th edition, 1994.