Simulating Facial Surgery Using Finite Element Models

R. M. Koch*, M. H. Gross*, F. R. Carls†
D. F. von Büren*, G. Fankhauser*, Y. I. H. Parish*

*Swiss Federal Institute of Technology (ETH), Zürich
† Dept. of Maxillofacial Surgery, University Hospital, Zürich

ABSTRACT

This paper describes a prototype system for surgical planning and prediction of human facial shape after craniofacial and maxillofacial surgery for patients with facial deformities. For this purpose it combines, unifies, and extends various methods from geometric modeling, finite element analysis, and image processing to render highly realistic 3D images of the post surgical situation. The basic concept of the system is to join advanced geometric modeling and animation systems such as *Alias* with a special purpose finite element model of the human face developed under *AVS*. In contrast to existing facial models we acquire facial surface and soft tissue data both from photogrammetric and CT scans of the individual. After initial data preprocessing, reconstruction, and registration, a finite element model of the facial surface and soft tissue is provided which is based on triangular finite elements. Stiffness parameters of the soft tissue are computed using segmentations of the underlying CT data. All interactive procedures such as bone and soft tissue repositioning are performed under the guidance of the modeling system which feeds the processed geometry into the FEM solver. The resulting shape is generated from minimizing the global energy of the surface under the presence of external forces. Photorealistic pictures are obtained from rendering the facial surface with the advanced animation system on which this prototype is built.

Although we do not claim any of the presented algorithms themselves to be new, the synthesis of several methods offers a new facial model quality. Our concept is a significant extension to existing ones and, due to its versatility, can be employed in different applications such as facial animation, facial reconstruction, or the simulation of aging. We illustrate features of our system with some examples from the Visible Human Data Set™

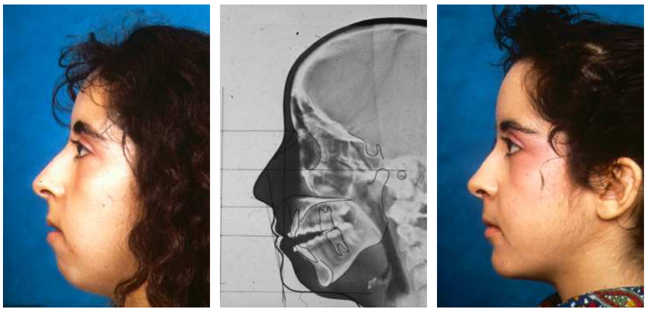
CR Descriptors: I.3.5 [**Computational Geometry and Object Modeling**]: Physically Based Modeling; I.3.7 [**Three-Dimensional Graphics and Realism**]; I.4.6 [**Segmentation**]: Edge and Feature Detection - Pixel Classification; I.6.3 [**Applications**];

Additional Keywords and Phrases: Finite Element Method, Facial Surgery Simulation, Facial Modeling, Data Reconstruction.

INTRODUCTION

Background and Previous Work

Since the human face plays a key role in interpersonal relationships, prediction of post-surgical morphology and appearance of human faces for patients with facial deformities is a critical issue in facial surgery. Even very subtle malformations of facial proportions can strongly affect the appearance of a face and determine on aesthetic aspects such as individual beauty [4]. In addition, many procedures in maxillofacial surgery lead to a change in the facial morphology of the patient, e.g. cutting of the jaw bones (osteotomies) and bringing them into a desired position. Maxillofacial surgery also consists of the resection of malformed parts of the face or tumors and the reconstruction of missing or malformed structures. Furthermore, the treatment of patients with facial fractures influences facial shape and surface. Therefore, surgeons often face the problem of predicting a fair facial surface before the actual surgery is carried out. Figure 1a and 1c illustrate a typical malformation of a young girl's face and its correction by surgery. Typically, the maxillofacial surgeon (sometimes with the help of a medical artist) draws the patients predicted profile to give at least a 2D appearance of the future face. A lateral X-ray with the actual and predicted soft tissue profiles is presented in Figure 1b. It is clear that both surgeons and their patients have a strong demand for a method which enables them to compute highly realistic 3D pictures of the post surgical shape. Any computation should be based on data available, or at least easy to obtain from the patient, e.g. CT scans. More specifically, with the development of advanced range scanners [15] the idea was born to combine both volume and surface data to build a physically based facial model. Obviously, the model has to capture the most important anatomical and mechanical parameters of the face. Furthermore, it should allow interactive manipulations and the prediction of resulting facial shape.



a) b) c)
Figure 1: Example of a facial disharmony resulting from a craniofacial disorder and its correction. a) Presurgical facial shape contour (profile). b) Lateral X-ray image of the skull with actual and planned new soft tissue profiles (2D). c) Postsurgical appearance after maxillofacial procedures. (Data source: Courtesy Prof. H. F. Sailer, Dept. of Maxillofacial Surgery, University Hospital Zürich, Switzerland)

It is straightforward to investigate computer graphics literature where facial modeling has a long tradition. Early works such as [12] restricted themselves to pure geometric deformations, mostly carried out directly on parametric surface models. However, with physically based modeling paradigms, more realistic facial models based on mass spring systems feature a linear approximation of the facial surface. The particle based approach is also used to govern control vertices of parametric B-spline models and thus provides continuous shape descriptions. In particular, the works of Terzopoulos e.a. [10, 16] extend the prior approaches in many significant ways: Anatomy is incorporated and an elaborate model of facial skin layers and muscles is proposed which is essential to compute realistic facial appearance.

In facial animation, computational complexity is a fundamental constraint, and real-time performance is often much more important than a highly accurate facial shape in terms of a still image.

*Swiss Federal Institute of Technology, Institute for Information Systems, Department of Computer Science, ETH Zentrum, CH-8092 Zürich, Switzerland. {grossm, koch}@inf.ethz.ch

To conclude, the methods mentioned above are based on strategies which exclude them from immediate use in facial surgery:

- In most cases facial models are adapted from a template face instead of generated directly from the underlying data set.
- The elastomechanics of the surface is only approximated by spring-mass systems which can be considered linear and hence C^0-continuous finite elements.
- Unsatisfactory approximations of the patient's skull surface were used which are essential for surgical applications.
- Although [10] proposes a very elaborate and elegant model, the underlying facial tissue is still template based and does not account for individual variations of the different layers and tissue types.

Our Approach

In the following paper we present a human facial model that enables prediction of facial deformations after surgery. In contrast to prior approaches, our goal is to approximate the physics as closely as possible and to accept higher computational costs. Therefore, we extend the models described earlier in 3 ways:

1. We employ a non-linear, globally C^1 continuous finite element model of the facial surface which is based on triangular polynomial shape functions similar to the modeling paradigm proposed by [2]. Increasing the number of finite elements results in a facial surface that tends to be C^3 continuous.
2. We compute the external forces of the model by connecting the surface with nodal springs to the skull. The individual spring stiffnesses are computed by means of 3D line integration through the CT data.
3. We build our model on top of commercially available modeling, visualization, and animation systems which support interactive geometric manipulation and rendering.

We organize our paper as follows: First, we give an overview of the system and describe the general set-up as a combination of commercial tools and individual components. In section 2 we elaborate on the different preprocessing steps to build the facial model. They range from geometry extraction and adaptive surface meshing to the computation of spring stiffnesses. Section 3 reviews the finite element model we employ. Particular emphasis is given to triangular patches. Section 4 illuminates the processing steps necessary to initialize the FEM model and perform computational surgery. Finally, we demonstrate the proposed system with experimental results obtained from the Visible Human Data Set™(VHD) [11].

1 System Overview

This section expands on the different procedures, data processing steps, and general set-up of our simulation system illustrated in the chart of Figure 2.

Our data sources consist of laser range scans, CT data or, for the VHD, photo slices. Our system combines a higher order polynomial FEM surface model with a mesh of springs attached to the skull. In order to generate a model for an individual patient, the following processing steps are necessary:

First of all, an initial facial surface is extracted from the data sources and an adaptive surface mesh computed for further processing. The mesh of springs connecting the surface and the skull requires that the skull has the same mesh topology as the skin. Therefore, we project the skin vertices down to the skull. One of the major advantages of our model is that it derives stiffness from the underlying volume data set, which makes a segmentation procedure necessary. All surgical planning is performed interactively on the precomputed skull model. Further interaction is required to define boundary conditions and the stretching and bending tensors crucial to achieve robust results. The FEM pipeline is comprised of computations for local and global stiffness matrices, preloading, assembling, solving and disassembling to obtain the new shape. Finally, the facial surface is rendered using *Alias*.

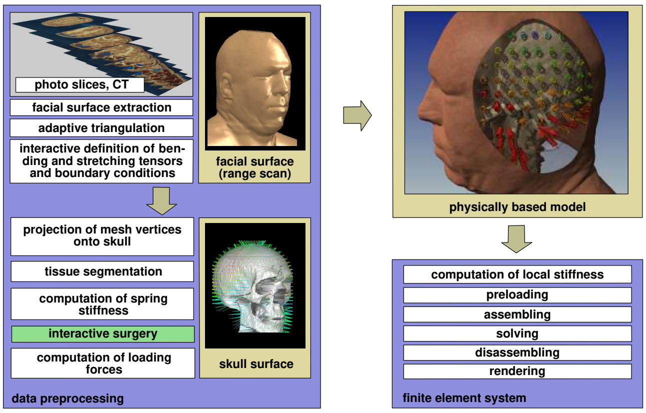


Figure 2: Illustration of data flow, procedures, and system set-up for maxillofacial surgery planning.

2 Data Acquisition and Preprocessing

The following section explains the different preprocessing steps essential for generating an initial facial model of the patient. Once again, most of the methods we employ are part of the standard repertoire of graphics and imaging algorithms, and hence we do not describe them all in detail here.

2.1 Facial Surface and Volume Data

As stated earlier, our system targets facial surgery simulation based on laser range surface data and CT volume data. Typical laser scans are often recorded by Cyberware scanners. Because scanning time is a critical issue due to motion artifacts, faster devices such as the one presented in [15] have been developed. In our experimental set-up we use the VHD as a reference. Figures 3a and 3b show examples of transaxial color photos and CT scans which have been scaled, clipped, and registered with respect to reference points. In order to extract the surface from the photo slices, we simulate a range scan procedure as described below.

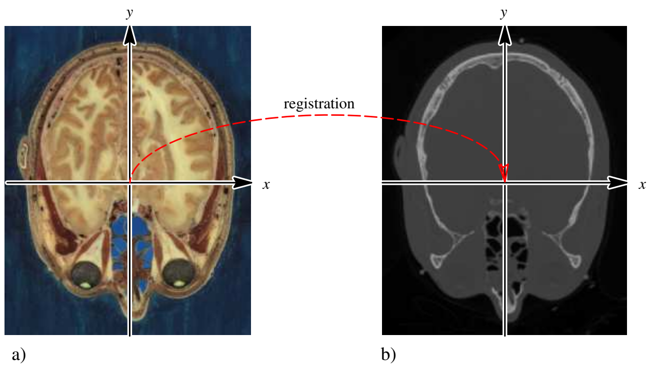


Figure 3: Volume data sets. a) Photo slice of the VHD. b) Corresponding CT slice. (Data source: Visible Human Project, Courtesy National Library of Medicine)

2.2 Extraction of the Facial Surface and Skull

The range scan simulation is accomplished by a two pass procedure, as shown in Figure 4. The first pass, which performs an edge detection using the Canny operator [1], is applied to the red channel of Figure 3a. Figure 4a shows the edge detection results. In the second pass, we intersect the image with rays through the center with a given angular resolution $\Delta\phi$. The respective range value is computed in accordance with the first intersection of the ray with the Canny edges. The result of this procedure is depicted in Figure 4b. It is clear that the surface computed in this way can be transformed into a cylindrical coordinate system without obtaining degenerated triangles. This surface coordinate system is initially applied to define tangential vector orientation used during the FEM processing step and is visualized as a grid mapped onto the surface. The parametric coordinates are denoted by (u, v). The same procedure holds for the extraction of the skull from the CT data set. Bone tissue produces high intensity values in CT data and can be enhanced with windowing operations. The final skull shape is shown in Figure 4c.

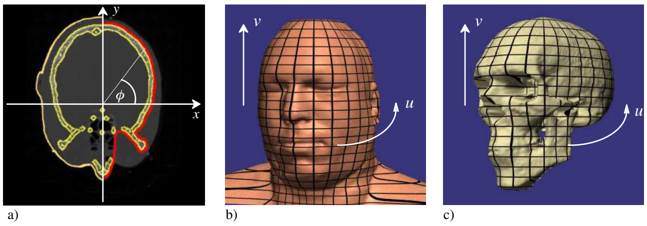


Figure 4: Facial surface and skull data extraction. a) Edges detected from a CT slice using the Canny edge detector (yellow edges) and simulated range scans for skull (red) and skin (skin-colored). b) Facial surface and underlying (u, v) coordinate system. c) Skull surface sharing the same coordinate system.

The above procedure is vulnerable to some minor artifacts due to self occlusion (e.g. nostrils, ears), but it successfully avoids the generation of non-manifold shapes as an isosurface tool might produce [14]. The resulting skull is sufficiently shaped to perform facial surgery.

2.3 Adaptive Meshing

The surface model is based on a finite element surface defined over an arbitrary triangulation of the parameter domain. Therefore, a trade-off has to be found between computational costs and precision. Since the computing time depends mostly on the number of triangular patches, a good adaptive triangulation of the parameter domain is an essential preprocessing step. There are a wide range of meshing and triangulation schemes available in the literature, like Delaunay, greedy meshing, or wavelets [5]. We decided to use a slightly modified version of the method of [13] for thinning and retiling. The mesh has to follow the 2D parameter domain and additional effort has to be spent at its boundaries. In our case, we start with a given trivial mesh directly obtained from the laser range scan as presented in Figure 5a. It is thinned out up to a given quality threshold, as visualized in Figure 5c. A mesh comparable to Figure 5b is used for all results presented in this paper. The final bust of the VHD was generated from the original data set by subsampling by 2. The initial mesh consisted of 69k triangles and was reduced to 39k.

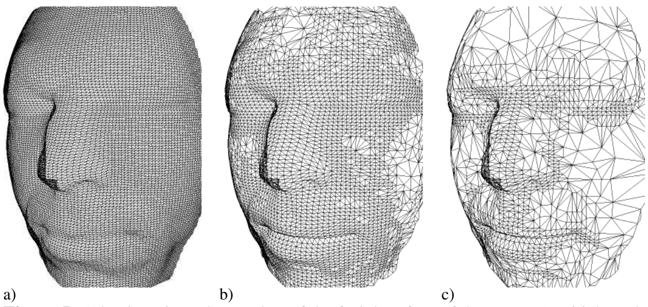


Figure 5: Adaptive triangular meshes of the facial surface of the VHD. a) Initial mesh generated by an equally spaced subsampling by $2^{0.5}$ (21k triangles). b) Adaptive mesh with 7k triangles after subsampling by 2. c) Adaptive mesh consisting of 2.5k triangles.

In order to remove high frequency spikes resulting from gaps in the Canny edges and from wire ropes in the VHD, additional Gaussian low pass filtering was performed locally.

2.4 Registration and Segmentation

Registration of facial surface and CT data is fundamental to the task of modeling the facial tissue. Although many automatic matching methods have been proposed in the computer vision literature [8], fully automatic techniques are still prone to errors. In the general case of laser range scans and CTs from different sources we carry out the registration manually by setting landmarks [6]. However, in the experimental set-up with the VHD, the facial surface is computed directly on the volume data set (see section 2.2) and registration is accomplished inherently.

In order to compute a 3D physically based model of the individual's facial tissue, it is necessary to provide detailed information on the different tissue types which have different physical parameters. [10] proposes an elaborate 5-layered soft tissue model with biphase springs, but individual compositions of the patient's facial tissue are not considered. For this reason, we base the computation of the spring stiffness on a segmentation of the underlying volume data. In the general case of CT data sets, much work has been done to provide appropriate tissue extraction, but intensity-based methods [3] still do a good job. In this case, the intensity range is segmented into different tissue types which tend to be Gaussian distributed, as shown in Figure 6a. The resulting segmentation is illustrated in Figure 6b. In the special case of photo slices, more elaborate color clustering methods can be applied, such as in Figure 6c.

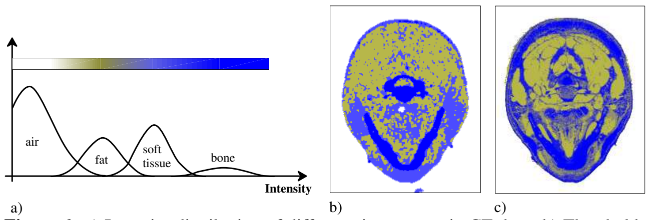


Figure 6: a) Intensity distribution of different tissue types in CT data. b) Threshold-based segmentation of a CT slice. c) Color-based segmentation of a photo slice.

For all results presented in this paper we have assigned the stiffness values of Table 1 to the different tissue types. These values are proportional to the corresponding spring stiffness values. Only tissue between skin and skull is considered.

Table 1: Spring stiffness parameters.

tissue type	color (red, green, blue)	stiffness
bone	(extracted from CT)	∞
skin	151, 125, 90	200.0
muscle	80, 40, 35	100.0
fat dark	165, 140, 100	10.0
fat bright	190, 160, 110	20.0

2.5 Spring Mesh Topology

Every vertex $\mathbf{x}_s^i$ of the facial surface mesh belongs to a vertex $\mathbf{x}_b^i$ on the skull. This relation enables us to construct a mesh of springs between skin and skull. The attachment point on the skull $\mathbf{x}_b^i$ is computed by tracing the surface normal $\mathbf{n}_i$ of the skin through the tissue in the direction to the skull. $\mathbf{n}_i$ is approximated by averaging the normals of all adjacent triangles. The skull intersection point $\mathbf{x}_b^i$ is found by thresholding the CT data set registered to the facial surface. We distinguish between two types of springs:

Main springs connect the skin vertices $\mathbf{x}_s^i$ with the skull vertices $\mathbf{x}_b^i$ on the bone. They are normal to the skin surface.

Strut springs connect a skin vertex $\mathbf{x}_s^i$ with the neighbors $\mathbf{x}_b^{neighbor(i,j)}$ of the related skull vertex $\mathbf{x}_b^i$.

Figure 7 illustrates this relationship. We do not use biphase springs

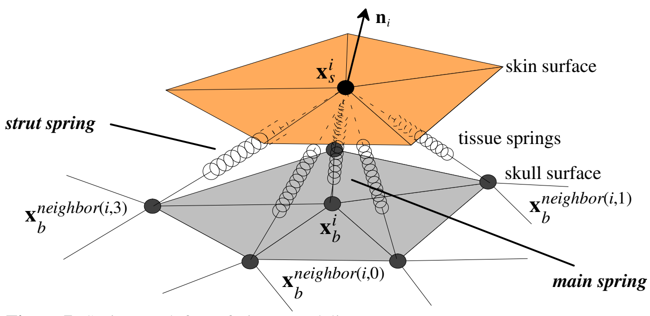


Figure 7: Spring mesh for soft tissue modeling.

since we are interested in the stationary shape, rather than dynamic behavior. Our tissue model assumes the skin directly connected to the underlying bone. There is no sliding between tissue and bone surface and none between different tissue types. Any extension of the model towards sliding and non-sliding tissue parts would require additional segmentations and is a task for future work.

2.6 Computation of the Spring Stiffness from CT

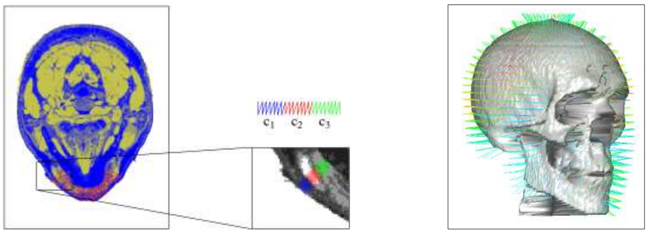
To compute the spring stiffnesses according to the segmentation presented in section 2.4, we have to consider that a spring penetrates different types of facial tissue on its way from the skin surface to the skull attachment point. We cope with this problem by decomposing the spring into a set of smaller springs which are connected in series.

Let $c(\mathbf{x})$ be the tissue stiffness as a function of the spatial parameter $\mathbf{x}$, and let $\mathbf{x}_s^i$ and $\mathbf{x}_b^i$ be the parameter values for the skin and bone attachment points of one individual spring k. The overall stiffness c_k of the resulting spring is computed by line integration from the surface node to the skull attachment point.

$$\frac{1}{c_k} = \int_{\mathbf{x}_s^i}^{\mathbf{x}_b^i} \frac{1}{c(\mathbf{x})} d\mathbf{x} \tag{1}$$

For a discrete set of L springs with individual lengths $l_k^1, \ldots, l_k^L$ and stiffnesses $c_k^1, \ldots, c_k^L$ connected in series, as in Figures 8a and b, the upper integral collapses to

$$\frac{1}{c_k} = \frac{1}{l_k} \sum_{h=1}^{L} \frac{l_k^h}{c_k^h} \qquad \text{with } \; l_k = \sum_{h=1}^{L} l_k^h \tag{2}$$



a) b) c)

Figure 8: a) Springs connecting the facial surface nodes with the skull. b) Decomposition into smaller springs connected in series as they penetrate different tissue types. c) Distribution of spring stiffnesses as pseudo-colored vectors.

This summation starts from the skin surface and runs along the surface normal to the skull attachment point. If the normal does not intersect the skull, a maximum distance from the surface is assumed and the spring stiffness is set to zero. The skull attachment points are also used in section 4.4 for calculation of the loading forces after bone repositioning.

As stated earlier, an adaptively triangulated surface mesh is applied. Hence, different springs k are affecting differently sized surface areas Δ_k. It is recommended to solve this problem by additional weighting of either the stiffness value c_k or the resulting force with Δ_k. See also section 4.4.

Figure 8c illustrates the distribution of spring stiffnesses for the VHD using pseudo-colored vectors.

3 The Finite Element Model

This section discusses the finite element surface model we use to represent the facial surface. Finite element modeling is a fundamental engineering method with many different types of elements proposed in the past. We focus on the construction of globally C^1 continuous shape functions over triangles. Global continuity is required for smooth representation and rendering of the facial shape. Furthermore, the triangular approach is topologically more flexible and fixes irregular meshes of the parameter domain. C^1 triangular finite elements have first been proposed by [17] and used by [2] to develop physically based modeling paradigms.

3.1 Variational Surface Modeling

The mathematical description of deformable surface models is well studied and can be considered a variational approach. The resulting shape $\mathbf{w}(u,v)$ is computed by minimizing the overall energy E_{surf} under the presence of external and internal loading forces. The shape's internal deformation energy E_{def} depends on its natural resistance against bending and stretching. α and β denote the respective tensors defining the individual material properties. The surface energy has to be computed over the parameter domain $\mathbf{\Omega}$. For convenience we follow the notation of [17] and [2].

$$E_{def} = \int_{\mathbf{\Omega}} (\alpha \, \mathbf{stretching} + \beta \, \mathbf{bending}) \, d\mathbf{\Omega} \tag{3}$$

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{12} & \alpha_{22} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{11} & & \\ & \beta_{22} & \\ & & \beta_{12} \end{bmatrix}$$

More specifically, let $\mathbf{f}(u,v)$ be the external forces applied to a surface $\mathbf{w}(u,v)$, and let $\mathbf{w}_u$, $\mathbf{w}_{uu}$, $\mathbf{w}_v$, $\mathbf{w}_{vv}$ be the first and second order derivatives in the principal directions. We can reformulate (3) to

$$E_{surf} = \tag{4}$$

$$\int_{\mathbf{\Omega}} \begin{matrix} \alpha_{11}\mathbf{w}_u^2 + \alpha_{12}\mathbf{w}_u\mathbf{w}_v + \alpha_{22}\mathbf{w}_v^2 + \\ \beta_{11}\mathbf{w}_{uu}^2 + 2\beta_{12}\mathbf{w}_{uv}^2 + \beta_{22}\mathbf{w}_{vv}^2 \end{matrix} \; - \; 2\mathbf{f}\mathbf{w} \; d\mathbf{\Omega}$$

The central idea underlying the finite element approach is to compute an approximate solution of the unknown shape $\mathbf{w}(u,v)$ by dividing $\mathbf{w}(u,v)$ into patches $\mathbf{w}_p(u,v)$ and using expansions with weighted sums of bases ϕ_l, which form the *finite elements*:

$$\mathbf{w}_p(u,v) \approx \sum_l \mathbf{a}_l \phi_l \tag{5}$$

where $(u,v) \in \mathbf{\Omega}_p$ is the local parameter space of a surface patch p.

These bases are defined over a finite domain $\mathbf{\Omega}_p$ provided by the initial discretization of the underlying parameter domain $\mathbf{\Omega}$. Thus, the problem of solving the upper functional is reduced to find appropriate weights $\mathbf{a}_l$. These weights represent the coordinates of the shape function in a functional space spanned by the ϕ_l.

Based on the upper relation, the functional (4) can be rewritten in matrix form so it collapses to

$$\mathbf{K}\mathbf{a} = \mathbf{F} \quad \begin{matrix} \mathbf{K}\text{: System global stiffness matrix} \\ \mathbf{F}\text{ : External load vector} \end{matrix} \tag{6}$$

This method allows the global stiffness matrix to be composed by assembling all local stiffness matrices of the different finite elements. $\mathbf{K}$ is usually a large, sparse matrix and sophisticated libraries for efficient handling are available [9].

This approach is fundamentally different from the finite difference method where the solution is only provided at discrete shape points.

Many different types of shape functions have been proposed in the past, depending on the job they have been tailored for. We face two essential requirements for our application:

1. Topologically flexible discretization of the parameter domain in terms of irregular triangular meshes.

2. Global C^1 continuity to ensure a smooth shape for further rendering.

This leads directly to C^1 shape functions over barycentric coordinates. Although these types of finite elements can be found in [17], there is no ready-to-use recipe for how to construct them. For this reason, we explain our construction scheme in detail below.

3.2 Globally C^1 Continuous Shape Functions

The globally C^1 continuous shape functions can be derived from the very popular $\mathbf{N}^9$ shape functions. Let r, s, t define a barycentric coordinate system with $r + s + t = 1$. The following 9 DOF polynomial elements $N_1, \ldots, N_9$ independently fix the nodal displacements and derivatives along the triangle edges, according to Figure 9. $N_1, \ldots, N_3$ are given by Hermite type polynomials (7). All other $N_4, \ldots, N_9$ can be computed by cyclic shifting of r, s, and t.

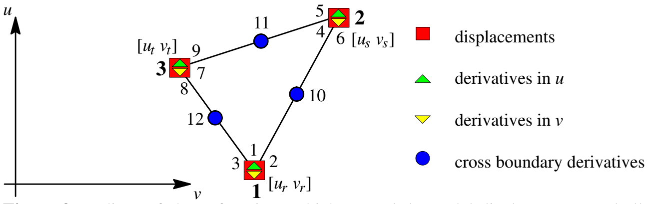


Figure 9: Indices of shape functions which control the nodal displacements and all derivatives.

$$
\begin{aligned}
N_1 &= r + r^2 s + r^2 t - r s^2 - r t^2 \\
N_2 &= r^2 s + \tfrac{1}{2} r s t \\
N_3 &= r^2 t + \tfrac{1}{2} r s t \\
N_{10} &= \tfrac{r^2 s^2 t(1+t)}{(r+t)(s+t)}
\end{aligned}
\tag{7}
$$

In order to achieve global C^1-continuity, we have to control the cross boundary derivative of two adjacent triangular patches without influencing the other DOFs. This can be accomplished by extending the initial set of shape functions with the polynomials N_{10}, N_{11}, and N_{12}. They only influence the cross boundary derivative of the respective edge. Figure 10 shows how N_1, N_2, N_3, and N_{10} affect the displacement and boundary derivatives of node r.

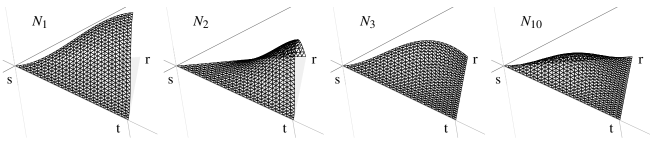


Figure 10: Illustration of the fundamental shape functions N_1, N_2, N_3, and N_{10} to control the displacements and derivatives.

Unfortunately, the cross boundary derivatives of our initial set of C^0 bases do not vanish. Therefore, it is necessary to correct these contributions. This correction is performed by evaluating the derivatives of the $N_1, \ldots, N_9$ shape functions normal to all triangle edges rs, st, and tr in the underlying global (u, v) parameter space. Let m_i^{rs} be the derivative of the shape function N_i normal to edge rs. We can define a modified set of shape functions $\bar{N}_i$ whose cross boundary derivatives vanish. This is done by correcting them with appropriate weighted sums of the functions $N_{10}, \ldots, N_{12}$.

$$
\bar{N}_i = N_i - 8\Delta \left(\frac{N_{10} m_i^{rs}}{len_{rs}} + \frac{N_{11} m_i^{st}}{len_{st}} + \frac{N_{12} m_i^{tr}}{len_{tr}} \right) \tag{8}
$$

The upper correction term is influenced by twice the triangular area Δ, by the individual lengths of the edges $(len_{rs}, len_{st}, len_{tr})$, and by a factor of 4 which is the inverse of the derivative values of N_{10}, N_{11}, and N_{12} at the edge midpoints.

Additional effort has to be spent to take into account the actual orientation of the triangular patch in the global surface coordinate system. The detailed description of the computation of m_i is beyond the scope of this paper but can be found in [2].

Now the definition of the final set of globally C^1 continuous shape functions ϕ_i is straightforward:

$$
\begin{aligned}
\phi_1 &= \bar{N}_1 & \phi_7 &= \bar{N}_7 \\
\phi_2 &= (c_3 \bar{N}_2 - c_2 \bar{N}_3) 2\Delta & \phi_8 &= (c_2 \bar{N}_8 - c_1 \bar{N}_9) 2\Delta \\
\phi_3 &= (-b_3 \bar{N}_2 + b_2 \bar{N}_3) 2\Delta & \phi_9 &= (-b_2 \bar{N}_8 + b_1 \bar{N}_9) 2\Delta \\
\phi_4 &= \bar{N}_4 & \phi_{10} &= sig_{rs} \tfrac{r^2 s^2 t(1+t)}{(r+t)(s+t)} \tfrac{2\Delta}{len_{rs}} \\
\phi_5 &= (c_1 \bar{N}_5 - c_3 \bar{N}_6) 2\Delta & \phi_{11} &= sig_{st} \tfrac{s^2 t^2 r(1+r)}{(s+r)(t+r)} \tfrac{2\Delta}{len_{st}} \\
\phi_6 &= (-b_1 \bar{N}_5 - b_3 \bar{N}_6) 2\Delta & \phi_{12} &= sig_{tr} \tfrac{t^2 r^2 s(1+s)}{(t+s)(r+s)} \tfrac{2\Delta}{len_{tr}}
\end{aligned}
\tag{9}
$$

We require

$$
\begin{aligned}
a_1 &= u_s v_t - u_t v_s \\
b_1 &= v_s - v_t \\
c_1 &= u_t - u_s
\end{aligned}
\tag{10}
$$

and for the sig operator which is incorporated in $\phi_{10}, \ldots, \phi_{12}$

if $(u_r - u_s = 0)$ {
    if $(v_r - v_s > 0)$ $sig_{rs} := 1$**; else** $sig_{rs} := -1$**;**
}
else {
    if $(u_r - u_s > 0)$ $sig_{rs} := 1$**; else** $sig_{rs} := -1$**;**
}

and finally get all other a_i, b_i, c_i, sig_{st}, and sig_{tr} by cyclic shift of the indices.

For each edge shared by two triangular patches we have to find a unique definition of the sign of their normal direction. This is necessary for the assembling step in the finite element system and is taken into account during the transformation of the shape function from the local barycentric coordinate system into the global cartesian coordinates.

Now, a global convention for the cross boundary normal $\mathbf{n}$ can be defined via the dot product:

$$
\mathbf{n} := \begin{cases} \mathbf{n} & \text{for} \quad \mathbf{n} \cdot \mathbf{v} > 0 \\ \mathbf{u} & \text{for} \quad \mathbf{n} \cdot \mathbf{v} = 0 \\ -\mathbf{n} & \text{else} \end{cases} \tag{11}
$$

where $\mathbf{v}$ is the $\mathbf{v}$-axis of the surface coordinate system at the edge midpoint.

Figure 11 shows two adjacent patches e_1 and e_2. In order to keep the C^1 continuity across the common edge, the corresponding cross boundary normals $\mathbf{n}_j$ and $\mathbf{n}_k$ have to point in the same direction.

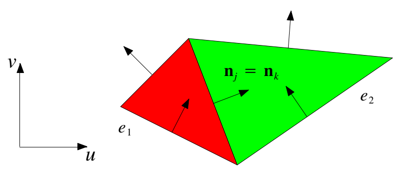


Figure 11: Orientation of cross boundary normals on two adjacent finite elements.

The set of shape functions $\phi_1, \ldots, \phi_{12}$ enables us to construct globally C^1 contiuous shapes over irregular triangular meshes. This is illustrated for a 3-patch surface with its center node displaced in the z-direction in Figures 12a-d for an increasing number of subdivisions. As the number of polygons increases, the approximation converges to a continuous shape.

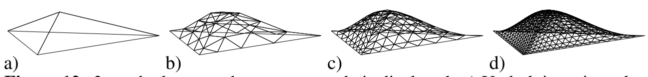


Figure 12: 3 patch element whose center node is displaced. a) Underlying triangular mesh. b), c), d) Approximations using 4, 8, and 16 edge subdivisions.

The final shape is determined by the weighted sum of the $\phi_1, \ldots, \phi_{12}$, as in (5). Due to the parametric approach, an individual ϕ_i has vector-related weights and the resulting weighting vector $\mathbf{a}$ has 36 entries:

$$
\mathbf{a} = [\mathbf{w}_{\phi_1}, \mathbf{w}_{\phi_2}, \ldots, \mathbf{w}_{\phi_8}, \mathbf{w}_{\phi_9}, \mathbf{w}_{\phi_{10}}, \mathbf{w}_{\phi_{11}}, \mathbf{w}_{\phi_{12}}]^T \tag{12}
$$

3.3 Computation of Local and Global Stiffness

As explained earlier, the central idea of the finite element approach is to approximate the global solution of the energy functional by piecewise shapes which are represented by linear combinations of the shape functions. Let $\mathbf{a}_e$ be the weight vector of a single finite element e and $\Phi_e = (\phi_1, \ldots, \phi_{12})^T$ be the corresponding vector of the shape functions. We obtain the fractional energy of an individual patch $\mathbf{E}_e$ by

$$
\begin{aligned}
\mathbf{E}_e = & \int\limits_{\mathbf{\Omega}_e} \mathbf{a}_e^T \left(\Phi_{e,str}^T \alpha \Phi_{e,str} + \Phi_{e,ben}^T \beta \Phi_{e,ben} \right) \mathbf{a}_e \, d\mathbf{\Omega}_e \\
& - 2 \int\limits_{\mathbf{\Omega}_e} \mathbf{f}^T \mathbf{\Phi}_e \mathbf{a}_e \, d\mathbf{\Omega}_e
\end{aligned}
\tag{13}
$$

Since $\mathbf{a}_e$ is constant over e, we are able to reformulate (13) as

$$\mathbf{E}_e = \mathbf{a}_e^T \mathbf{K}_e \mathbf{a}_e - 2\mathbf{F}_e \mathbf{a}_e \tag{14}$$

where $\mathbf{K}_e$ und $\mathbf{F}_e$ represent the local patch stiffness matrix and the external load vector

$$\mathbf{K}_e = \int_{\mathbf{\Omega}_e} \left(\mathbf{\Phi}_{e,str}^T \alpha \mathbf{\Phi}_{e,str} + \mathbf{\Phi}_{e,ben}^T \beta \mathbf{\Phi}_{e,ben} \right) \, d\mathbf{\Omega}_e \tag{15}$$

$$\mathbf{F}_e = \int_{\mathbf{\Omega}_e} \mathbf{f}^T \mathbf{\Phi}_e \, d\mathbf{\Omega}_e \tag{16}$$

In order to compute $\mathbf{K}_e$, it is neccessary to evaluate the first and second order derivatives $\mathbf{\Phi}_{e,str}$ and $\mathbf{\Phi}_{e,ben}$ of the $\mathbf{\Phi}_e$ related to the barycentric coordinate system. This is accomplished by a set of operators which are explained in [2].

Although it is possible to provide analytic descriptions of these derivatives they end up in very complex expressions. Hence, we use finite differences to approximate them in our current implementation. The outer integral of the stiffness matrix is calculated by Gaussian quadrature.

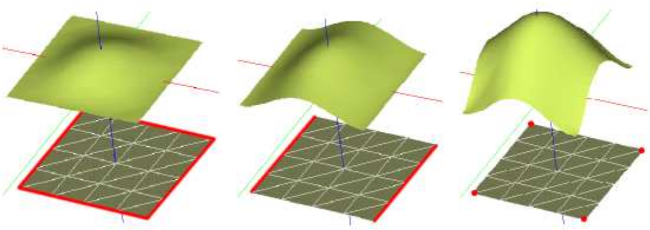
The last step of the finite element approach is to assemble the local stiffness matrices and force vectors into a global system of equations, according to (6). Therefore, all local matrices and vectors are superimposed in accordance to their topological relations.

4 Computational Surgery

This section explains additional computations, constraints, parameters, and boundary conditions neccessary to accomplish facial surgery simulations. A fundamental precondition is the definition of the boundary conditions of the finite element shape in terms of rigid and non-rigid nodes. Furthermore, we have to assign different stretching and bending tensors for each local finite element to compute the local stiffness matrices. The computation of the initial shape vector itself has to be performed in advance because it can be considered an initial equilibrium state of the shape. Nodal displacement forces applied onto the finite elements are due to the tissue springs attached to the skull.

4.1 Boundary Conditions

For all finite element approximations of differential equations, the boundary conditions are fundamental. Boundary conditions define a unique solution and diminish the overall degree of freedom. Figure 13 illustrates the influence of different boundary conditions on the final shape for a simple 32 patch surface. Although the same external force vector is applied, due to the different boundary conditions at the four boundaries (depicted in red), the resulting shapes are not equal. Note that the boundary conditions are set both for nodal displacements and for all derivatives.

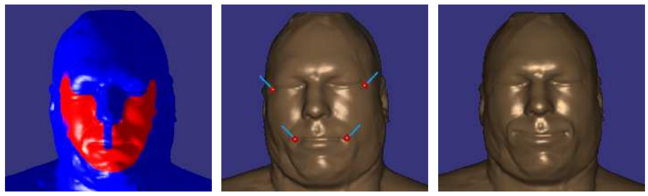


a) b) c)

Figure 13: Different boundary conditions (red) and their influence on a simple shape under the presence of the same external load vector. a) All edges fixed. b) Vertical edges fixed. c) Corner vertices only fixed.

The same situation holds for the facial skin surface. A careful definition of the boundary conditions is critical to a satisfying solution and helps to enhance computational performance by lowering the dimensionality of the global stiffness matrix. Since in facial surgery most procedures affect only parts of the facial shape, we explicitly define displacement conditions. A distinction between *rigid* and *non-rigid* nodes is proposed, i.e. only non-rigid nodes can be displaced during the FEM computation. This is very effective because even in very complicated cases, backside, neck, and hair do not change dramatically in shape.

For users' convenience we define the whole set of boundary conditions in terms of binary texture maps on the facial shape, as shown in Figure 14a. The red regions are non-rigid, the blue ones are rigid. The texture map itself can be generated with an interactive paint program, for instance *Alias Paint*. In Figure 14c a typical deformation of the facial shape is presented in accordance with the boundary conditions set in Figure 14a and the nodal displacement forces indicated as blue lines in Figure 14b.



a) b) c)

Figure 14: a) Boundary conditions represented as a texture map onto the range data (blue: rigid nodes, red: non-rigid nodes). b) Interactive definition of nodal forces. c) Deformation of the facial shape in accordance with the boundary conditions on the left.

4.2 Stretching and Bending Parameters

Similar to the spring stiffness, the stretching and bending tensors may change their values for different regions of the face. More specifically, facial skin stiffness is a function of age, sex, weight, and other parameters [4]. Higher values lead to higher internal energy of a deformed surface patch, thus giving rigid parts of the face high α and β values. Changing the stretching and bending parameters influences the size of the deformed facial region and the displacement of individual surface nodes if a loading force is applied.

An interactive paint procedure is also used to assign stretching and bending parameters of different regions of the face. A well-suited facial segmentation algorithm for the automatic setting of facial stretching and bending parameters can be found in [6]. For all computations in this paper, the parameters were set constant over the surface of one triangular patch. Typical settings are $\alpha = 0.01$ for stretching, and $\beta = 0.02$ for the bending factor. Higher values guarantee appropriate deformations of facial regions with underlying cartilage (ears, nose). Figure 15 illustrates the variation of the surface parameters color-coded for the VHD. Darker colors indicate positive deviations and brighter colors negative deviations from the values above.

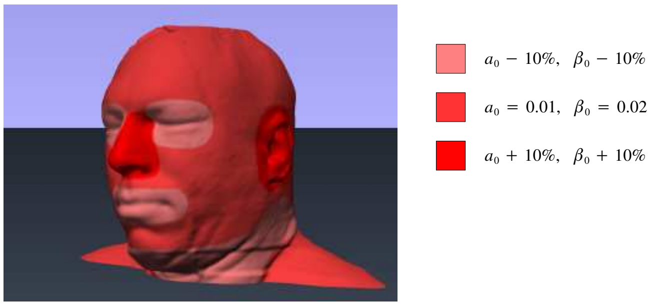


Figure 15: Variation of the stretching and bending tensors as a texture map.

4.3 Initialization

A fundamental task during a FEM modeling step of the kind we propose here is to compute the initial shape of the face from the input data. More specifically, we have to determine the initial weight vector $\mathbf{a}_{init}$ which controls the contribution of each shape function to the overall facial shape based on adaptive triangular meshes like the one in Figure 5. Conversely, once the initial weight vector is derived from the underlying data set, the FEM approach provides us with a set of *virtual nodal preloading vectors* $\mathbf{F}_{init}$, as given in (17). Although this initial preloading is inherent to the approach, it can be considered an external energy which forces the deformable surface into the initial presurgical facial shape.

$$\mathbf{F}_{init} = \mathbf{K}\mathbf{a}_{init} \tag{17}$$

In other words, the initialization of our model is a two pass process:

1. Estimation of the $\mathbf{a}_{init}$ weight vector from the initial facial surface.
2. Computation of the preload vector $\mathbf{F}_{init}$.

The estimation of $\mathbf{a}_{init}$ can be carried out as follows: The displacement vectors $\mathbf{w}_{\phi_1}$, $\mathbf{w}_{\phi_4}$, and $\mathbf{w}_{\phi_7}$ are given immediately from the vertex positions of the triangular mesh. Any nodal derivative vector in (u, v) direction $\mathbf{w}_{\phi_2}$, $\mathbf{w}_{\phi_3}$, $\mathbf{w}_{\phi_5}$, $\mathbf{w}_{\phi_6}$, $\mathbf{w}_{\phi_8}$, and $\mathbf{w}_{\phi_9}$ as well as any cross boundary derivative vector $\mathbf{w}_{\phi_{10}}$, $\mathbf{w}_{\phi_{11}}$, and $\mathbf{w}_{\phi_{12}}$ at the midpoints of the triangle edges are computed using the approximated surface normals $\mathbf{n}_{uv}$ at the coordinates (u, v). $\mathbf{n}_{uv}$ is computed by averaging the normals of adjacent triangles. For a given normal vector $\mathbf{n}_{uv}$, the slope vectors in u and v direction can be approximated as follows:

$$\begin{aligned} \frac{\partial \mathbf{w}(u,v)}{\partial u} &\approx \frac{\mathbf{a}_{cyl}\times\mathbf{n}_{uv}}{\left\|\mathbf{a}_{cyl}\times\mathbf{n}_{uv}\right\|} \\ \frac{\partial \mathbf{w}(u,v)}{\partial v} &\approx \frac{\frac{\partial \mathbf{w}(u,v)}{\partial u}\times\mathbf{n}_{uv}}{\left\|\frac{\partial \mathbf{w}(u,v)}{\partial u}\times\mathbf{n}_{uv}\right\|} \end{aligned} \tag{18}$$

where $\mathbf{a}_{cyl}$ is the rotation axis of the underlying cylindrical coordinate system. The cross boundary slopes at the edge midpoints are derived as (19).

$$\begin{aligned} \mathbf{w}_{\phi_{10}} &\approx sig\frac{\mathbf{e}_{rs}\times\mathbf{n}_{uv}}{\|\mathbf{e}_{rs}\times\mathbf{n}_{uv}\|} \\ \mathbf{w}_{\phi_{11}} &\approx sig\frac{\mathbf{e}_{st}\times\mathbf{n}_{uv}}{\|\mathbf{e}_{st}\times\mathbf{n}_{uv}\|} \\ \mathbf{w}_{\phi_{12}} &\approx sig\frac{\mathbf{e}_{tr}\times\mathbf{n}_{uv}}{\|\mathbf{e}_{tr}\times\mathbf{n}_{uv}\|} \end{aligned} \tag{19}$$

where sig are the respective signs to guarantee condition (11), and $\mathbf{e}_{rs}$, $\mathbf{e}_{st}$, and $\mathbf{e}_{tr}$ are vectors pointing in the edge direction.

Figure 16a illustrates the estimated derivatives encoded as small lines of different colors around the nose and mouth. The resulting FEM shape is presented in Figure 16b, where the shape functions are subdivided by 2.

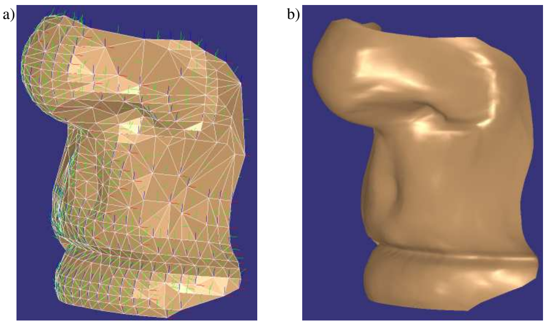


Figure 16: Triangulated part of the facial skin (nose). a) The nodal derivatives in **u**,**v** direction are represented in red and blue and the cross boundary derivatives are shown in green. b) Initial shape as a result of the precomputation step.

4.4 Repositioning of Bones: Computation of Nodal Loading Forces

We perform any surgical procedure on the skull as a geometric modeling step with *Alias*. The skull used for surgery is shown in Figure 17a. As stated earlier, the initial facial mesh is projected along the normals of every surface vertex onto the skull. This provides a mesh on the skull surface which is topologically identical to the skin mesh, as in Figure 17b. This skull has many artifacts arising from the underlying generation method. The jaw bone is marked in a different color and will be repositioned subsequently.

After surgery on the skull the translations $\mathbf{t}^i$ and $\mathbf{t}^{\mathrm{neighbor}(i)}$ of the spring-related skull attachment points at position $\mathbf{x}_b^i$ and $\mathbf{x}_b^{\mathrm{neighbor}(i)}$ have to be computed. This is performed by matching

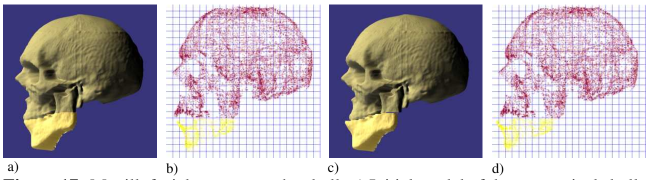


Figure 17: Maxillofacial surgery on the skull. a) Initial model of the presurgical skull. b) Skin vertices projected onto the skull. c) Final skull after repositioning of the jaw bone. d) Projected vertices for c).

the affected regions of the skull in Figure 17b with the model in Figure 17c. The result of this procedure is illustrated in Figure 17d.

Any deformation of the skull generates a force $\mathbf{f}_{def,i}$ on the surface node $\mathbf{x}_s^i$ which influences the energy equilibrium (13) between the internal energy of the surface and the preloading energy described in the former section. $\mathbf{f}_{def,i}$ is computed as follows:

$$\mathbf{f}_{def,i} = k_c k_i \sum_{j\in S_i} \frac{\mathbf{s}_{ij} - \mathbf{t}^j}{\|\mathbf{s}_{ij} - \mathbf{t}^j\|} c_{ij} \left(\|\mathbf{s}_{ij}\| - \|\mathbf{s}_{ij} - \mathbf{t}^j\| \right) \tag{20}$$

Vertex indices are equal for related skin and skull nodes.

$\mathbf{f}_{def,i}$	Deformation force at skin vertex i as a result of a deformation of the skull.
k_c	Force scaling constant.
k_i	Mesh dependent scaling factor.
S_i	Set of skull nodes for all springs attached at vertex i on the skin.
$\mathbf{s}_{ij}$	Spring connecting skin vertex i at position $\mathbf{x}_s^i$ with skull vertex j at position $\mathbf{x}_b^j$, i.e. $\mathbf{x}_s^i - \mathbf{x}_b^j$.
$\mathbf{t}^j$	Translation vector of the skull vertex j as a result of a surgery on the skull.
c_{ij}	Stiffness of spring $\mathbf{s}_{ij}$.

The mesh dependent scaling factor k_i is defined by

$$k_i = \sum_{j\in P_i} \Delta_j \tag{21}$$

where P_i is the set of all triangles containing node i, and Δ_j the area of the mesh triangle j. Figure 18 illustrates the upper relations for a particular node i.

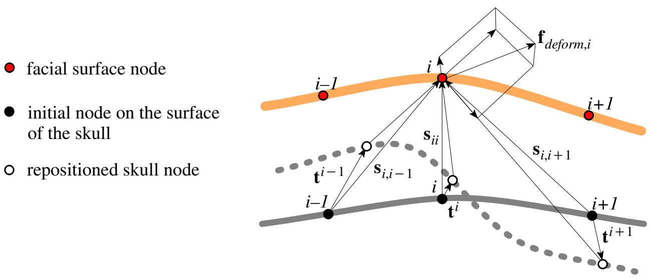


Figure 18: Computation of the nodal loading forces.

The deformation energy of the facial surface is derived from the nodal displacement forces as follows:

$$\mathbf{F}_{def} = \int_\Omega \mathbf{f}_{def}^T \mathbf{\Phi}\, d\mathbf{\Omega} \tag{22}$$

4.5 Solving the Global System

The external energy vector $\mathbf{F}_{ext}$ is computed by superimposing the nodal preloading energy $\mathbf{F}_{init}$ with the deformation energy $\mathbf{F}_{def}$.

The global energy equilibrium can be reformulated for the resulting shape vector $\mathbf{a}_{result}$ by

$$\mathbf{K}\mathbf{a}_{result} = \mathbf{F}_{init} + \mathbf{F}_{def} \tag{23}$$

It is recommended to carry out the simulation in small steps, including the recalculation of the patch stiffnesses of all patches affected in the last step.

5 Results

The goal was to predict the facial shape after standard procedures in craniomaxillo facial surgery. Figures 19a-c show the shapes of the skull and face before and after an osteotomy and advancement of the lower jaw bone. These pictures represent the corresponding deformations of skin in profile and frontal view. Obviously, the surgical procedure strongly affects the appearance of the face. This is even more striking in Figures 19d-f where an osteotomy and advancement of the upper jaw bone is presented. All computations are based on the methods described above. The pictures are rendered with *Alias*.

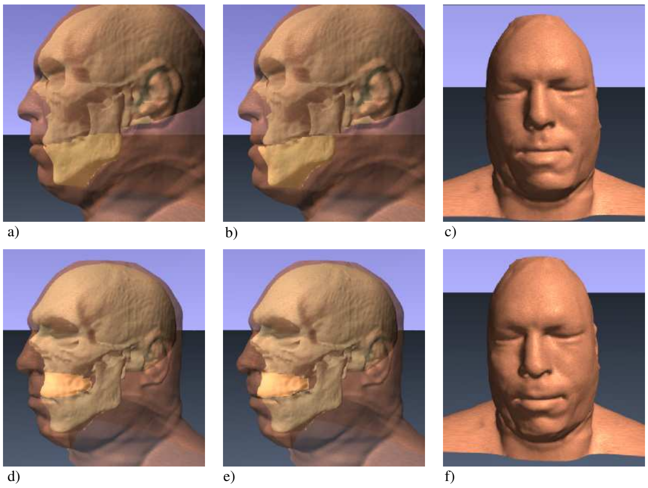


Figure 19: Results on the VHD. a), b), and c) Profiles and frontal view before and after advancement of the lower jaw. d), e), and f) Situation before and after surgery on the upper jaw.

The affected facial region of the first surgical procedure comprises about 3.1k triangles. The computation of the global stiffness matrix took approximately 17 minutes, the solving of the global equation system 11 minutes. The calculations were carried out on a *SGI Indigo*^{2}, R4400, 200MHz, 192MB.

6 Discussion and Future Work

We present a system which enables us to predict deformations of the facial shape after surgical procedures. Our prototype system uses commercial tools for geometric modeling and rendering. In contrast to existing approaches in physically based modeling of facial shapes, our approach is based on non-linear finite elements, and therefore provides a continuous approximation of the facial geometry. We consider our approach to be a significant extension in this field because we build the model on both laser range scans and CT volume data. This enables us to compute individual physical parameters. Some results achieved with our system are demonstrated using the VHD.

We are aware that some processing steps presented here are specific to the underlying data. Usually neither high resolution CT scans comparable to the VHD, nor photo slices are available. With future generations of MR and CT scanners, however, we expect that both facial surface and skull of the patient will be available at sufficient resolutions to make range scans obsolete.

Our future research is directed towards true 3D volume FEM systems based on tetrahedralizations. They would integrate both the current surface and volume spring model and allow incorporation of more anatomic features. In this case, the global C^1 constraint could be relaxed and replaced by local continuity properties. Further effort has to be spent on our current geometric model of skull and skin, which is based on polygons. It is clear that polysets perform worse than NURB representations and do not guarantee lossless data conversions between the FEM system and *Alias*. Other important topics are incompressible finite elements which can cope with the problem of wrinkles. The current rendering quality has to be enhanced with using texture maps and better reflection models for facial skin. Sophisticated models, such as in [7], have to be investigated for suitability. Finally, our goal is to carry out a case study with individual patients of different ages, genders and ethnic origins and to compare the computed results with those achieved by actual surgery. We expect also that an error quantification would help us to improve the model towards better performance.

7 Acknowledgement

The authors would like to thank Caroline Westort for her native-english review of the text.

References

[1] CANNY, J. F. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6 (1986), pp. 679–697.

[2] CELNIKER, G., AND GOSSARD, D. Deformable curve and surface finite elements for free-form shape design. In *Computer Graphics (SIGGRAPH '91 Proceedings)* (July 1991), ACM SIGGRAPH, T. W. Sederberg, Ed., vol. 25, pp. 257–266.

[3] DREBIN, R. A., CARPENTER, L., AND HANRAHAN, P. Volume rendering. In *Computer Graphics (SIGGRAPH '88 Proceedings)* (Aug. 1988), ACM SIGGRAPH, J. Dill, Ed., vol. 22, pp. 65–74.

[4] FARKAS, L. G. *Anthropometry of the Head and Face*, second ed. Raven Press, 1994.

[5] GROSS, M. H., GATTI, R., AND STAADT, O. Fast multiresolution surface meshing. In *Proceedings of IEEE Visualization '95* (1995), IEEE Computer Society Press, pp. 135–142.

[6] GROSS, M. H., AND KOCH, R. Visualization of multidimensional shape and texture features in laser range data using complex-valued Gabor wavelets. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (Mar. 1995), pp. 44–59. ISSN 1077-2626.

[7] HANRAHAN, P., AND KRUEGER, W. Reflection from layered surfaces due to subsurface scattering. In *Computer Graphics (SIGGRAPH '93 Proceedings)* (Aug. 1993), ACM SIGGRAPH, J. T. Kajiya, Ed., vol. 27, pp. 165–174.

[8] HARALICK, R. M., AND SHAPIRO, L. G. *Computer and Robot Vision*, vol. 1–3. Addison-Wesley, 1992.

[9] KUNDERT, K. S., AND SANGIOVANNI-VINCENTELLI, A. *A Sparse Linear Equation Solver*. Dept. of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1988.

[10] LEE, Y., TERZOPOULOS, D., AND WATERS, K. Realistic face modeling for animation. In *Computer Graphics (SIGGRAPH '95 Proceedings)* (Aug. 1995), ACM SIGGRAPH, R. Cook, Ed., vol. 29, pp. 55–62.

[11] NATIONAL LIBRARY OF MEDICINE. *The Visible Human Project*. http://www.nlm.nih.gov/extramural_research.dir/visible_human.html, 1995.

[12] PARKE, F. I. Parameterized models for facial animation. *IEEE Computer Graphics and Applications 2* (Nov. 1982), pp. 61–68.

[13] SCHROEDER, W. J., ZARGE, J. A., AND LORENSEN, W. E. Decimation of triangle meshes. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (Aug.1992), ACM SIGGRAPH, E. E. Catmull, Ed., pp. 65–70.

[14] VAN GELDER, A., AND WILHELMS, J. Topological considerations in isosurface generation. Tech. Rep. UCSC-CRL-94-31, Baskin Center for Computer Engineering and Information Sciences, University of California, Santa Cruz, 1994.

[15] VANNIER, M. W., PILGRAM, T., BHATIA, G., BRUNSDEN, B., AND COMMEAN, P. Facial surface scanner. *IEEE Computer Graphics and Applications 11*, 6 (Nov. 1991), pp. 72–80.

[16] WATERS, K. A muscle model for animating three-dimensional facial expression. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (July 1987), ACM SIGGRAPH, M. C. Stone, Ed., vol. 21, pp. 17–24.

[17] ZIENKIEWICZ, O. C. *The Finite Element Method*, fourth ed., vol. 1–2. McGraw-Hill, 1994.