

Head shop: Generating animated head models with anatomical structure

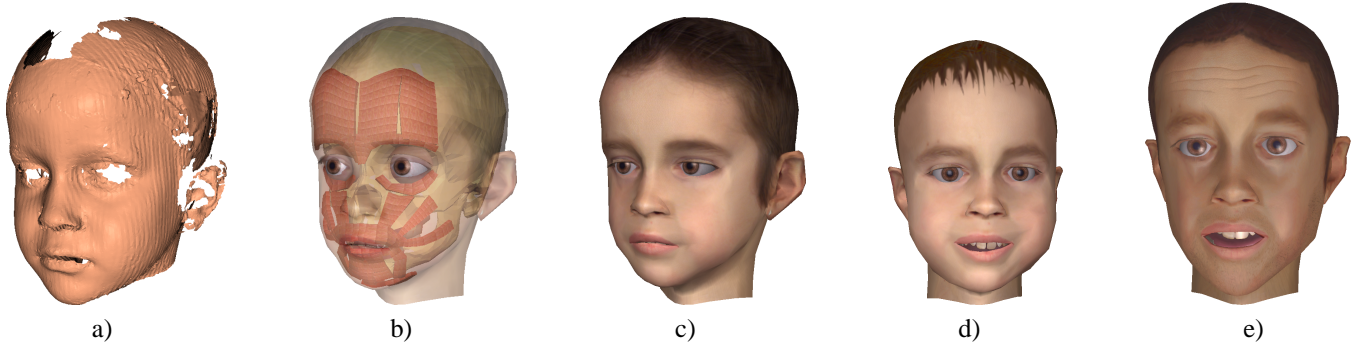
Kolja Kähler

Jörg Haber

Hitoshi Yamauchi

Hans-Peter Seidel

Max-Planck-Institut für Informatik, Saarbrücken



Head models generated from a range scan of a five year old boy: a) scan data; b) adapted, animatable head structure; c) textured; d) age changed to one year, smiling expression; e) age changed to 20 years, surprised expression.

Abstract

We present a versatile construction and deformation method for head models with anatomical structure, suitable for real-time physics-based facial animation. The model is equipped with landmark data on skin and skull, which allows us to deform the head in anthropometrically meaningful ways. On any deformed model, the underlying muscle and bone structure is adapted as well, such that the model remains completely animatable using the same muscle contraction parameters. We employ this general technique to fit a generic head model to imperfect scan data, and to simulate head growth from early childhood to adult age.

CR Categories: G.1.2 [Numerical Analysis]: Approximation—*approximation of surfaces, least squares approximation*; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*hierarchy and geometric transformations, physically based modeling*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*animation*.

Keywords: Biological Modeling, Deformations, Facial Animation, Geometric Modeling, Morphing, Physically Based Animation

1 Introduction

Recent advances in facial animation systems show the potential of physics-based approaches, where the anatomical structure of a human head is simulated, including skin, muscles, and skull [Waters and Frisbie 1995; Lee et al. 1995]. On current hardware, this kind of techniques can be used to create detailed, realistic animations in real-time. As the model becomes more complex, the assembly of the components becomes more complicated, thus giving rise to a strong interest in automated methods for head model construction. Once a model exists, it is often desirable to change some of its characteristics, or generally adapt it to another head geometry, while retaining full animation capabilities.

The more realistically the heads of existing human individuals can be reproduced in the computer, the more appealing is the use of anthropometric methods and data for analysis and modification of head geometry [DeCarlo et al. 1998]. Measurements of human

heads from different samples of the world population have been systematically collected over the past decades [Farkas 1994], resulting in a database of the predominant facial characteristics for individuals of different sex, age, and ethnicity. Anthropometric methods are usually based on *landmarks*, i.e. well-defined features on the face and – in the forensic sciences – also on the skull. In recent years, the topic of deformation of biological shapes described by such sets of landmark data has been treated by the emerging science of morphometrics [Bookstein 1997a].

We propose the use of anthropometric landmarks and an associated deformation technique based on thin-plate splines, arriving at a unified, elegant framework for a variety of tasks in facial modeling and animation. The added layer of abstraction over the implementation details of the structured head model allows for modification of the head geometry in terms of distance relations between facial features. Given a reference head model tagged with landmarks on the skin and bone layers, we automatically deform not only the outer skin geometry, but also the internal structure composed of muscles and skull. All head models derived from the reference head share the same set of animation parameters, i.e. muscles, enabling re-use of existing animation scripts.

The main contributions presented in this paper are:

- a general method to deform an animated head model with underlying anatomical structure, tagged with anthropometrically meaningful landmarks. The head model is suitable for real-time animation based on simulation of facial muscles and elastic skin properties.
- an algorithm to fit such a reference head model to even very poor scan data, using this deformation technique.
- a technique that utilizes the anthropometric measurements on the head model to simulate growth of a human head, employing the same deformation method, and resulting in animatable head models of an individual at different ages.

While we make extensive use of the body of knowledge and data collected for human faces, the general approach is applicable just as well to other classes of animated virtual creatures, provided a reference model with skull, muscles, and skin can be built.

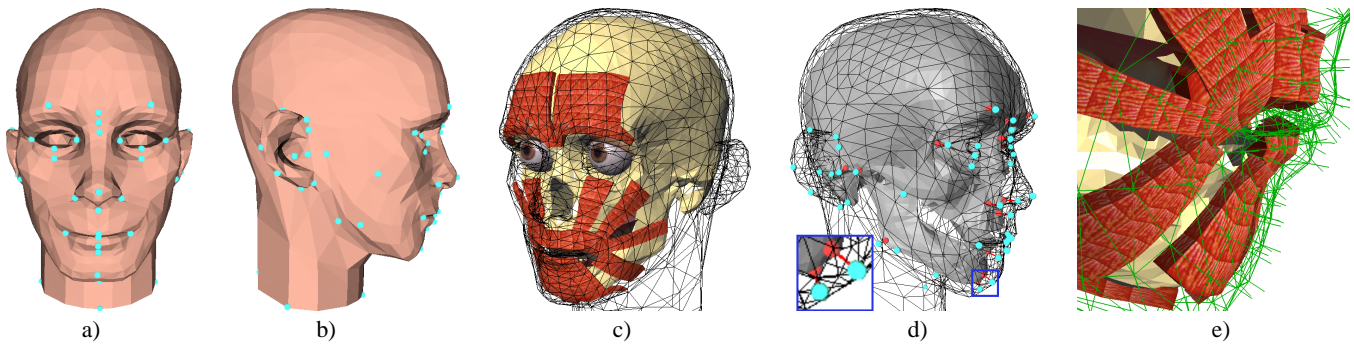


Figure 1: The reference head: a) head geometry with landmarks, front view; b) side view; c) skull and facial components; d) skull landmarks related to subset of skin landmarks; e) facial detail showing spring mesh connecting skin and muscles.

2 Previous and related work

Several facial animation systems use some approximation of layered anatomical structure. The idea of representing skin and muscles as separate entities was used by WATERS [1987], where muscle vectors and radial functions derived from linear and sphincter muscles specify deformations on a skin mesh. Also building on the idea of virtual muscles, physics-based approaches attempt to model the influence of muscle contraction onto the skin surface by approximating the biomechanical properties of skin. Typically, mass-spring or finite element networks are used for numerical simulation [Platt and Badler 1981; Lee et al. 1993; Koch et al. 1998]. TERZOPOULOS and WATERS [1990] automatically construct a layered model of the human face from an initial triangle mesh. The structure consists of three layers representing the muscle layer, dermis, and epidermis. The skull is approximated as an offset surface from the skin. This model was later simplified by LEE et al. [1995] for efficiency. Free-form deformations have been employed by CHADWICK et al. [1989] to shape the skin in a multi-layer model, which contains bones, muscles, fat tissue, and skin. SCHEEPERS et al. [1997] and WILHELMS and VAN GELDER [1997] introduced anatomy-based muscle models for animating humans and animals, focusing on the skeletal musculature. Skin tissue is represented only by an implicit surface with zero thickness [Wilhelms and Van Gelder 1997].

A variety of techniques exist to create a face model from images or scan data. In the method presented by LEE et al. [1995], animatable head models are constructed semi-automatically from range scans. A generic face mesh with embedded muscle vectors is adapted to range scans of human heads. This process relies on a planar parameterization of the range scans as delivered e.g. by the Cyberware digitizers. PIGHIN et al. [1998] interactively mark corresponding facial features in several photographs of an individual to deform a generic head model using radial basis functions. Animation is possible by capturing facial expressions in the process and blending between them. Employing a large database of several hundred scanned faces, BLANZ et al. [1999] are able to create a geometric head model from only a single photograph. The model has the same resolution as the range scans in the database and cannot be readily animated. CARR et al. [2001] also use radial basis functions to generate consistent meshes from incomplete scan data. In medical imaging, SZELISKI et al. [1996] minimize the distance between two surfaces obtained from volume scans of human heads by applying local free-form deformations [Sederberg and Parry 1986] and global polynomial deformations. The method does not require specification of corresponding features on the geometries.

A variational approach is presented by DECARLO et al. [1998] to create a range of static face models with realistic proportions. They use anthropometric measurements, which constrain the defor-

mation of a generic head model represented by a B-spline surface.

Recently, the transfer of animations between different head models on the geometric level has been proposed [Noh and Neumann 2001]. Surface correspondences are obtained by specification of corresponding point pairs on the models. Heuristics for automatic feature detection are presented which help to automate the process.

The work on aging in human faces has so far concentrated on the appearance of the skin, neglecting the considerable geometric changes that occur during growth. WU et al. [1999; 1994] focus on generation of expressive wrinkles and skin aging effects. Their muscle-driven face model incorporates viscoelastic properties of skin. LEE et al. [1999] reconstruct textured low polygon face models from photographs of the members of a family, simulating age changes by blending geometry and textures between young and old family members. Wrinkle patterns are generated semi-automatically by considering muscle fiber orientation and feature points on the face. LANITIS et al. [1999] present a statistical face model to isolate age variations in face images for age estimation and aging simulation. TIDDEMAN et al. [2001] use wavelet-based methods to identify salient features such as age wrinkles in prototype facial images, and apply them to other images to change the apparent age.

3 The reference head model

Our system builds on a prototype head model that has been designed for use in our physics-based animation system. The model encapsulates five major structural components, shown in Figure 1:

- a triangle mesh for the *skin surface*. The edges are aligned to facial features to reduce animation artifacts. The tessellation is adjusted to the deformability of the facial regions.
- a layer of *virtual muscles* to control the animation. The muscles consist of arrays of fibers which can contract in a linear or circular fashion. We have modeled 24 of the major muscles responsible for facial expressions and speech articulation.
- an embedded *skull*, including a rotatable mandible, to which skin and muscles attach. The skull is also represented as a triangle mesh and is only used during initialization of the structure, not during animation itself.
- a *mass-spring system* connecting skin, muscles, and skull. Basically, the edges and vertices of the skin surface mesh are converted to springs and point masses, respectively. More springs are added to connect to underlying components and to preserve skin tissue volume during animation.
- separately modeled components for *eyes, teeth, and tongue*.

The model structure has been designed manually, employing the muscle editing methods presented in [Kähler et al. 2001]. We have enhanced the described muscle model in several respects. The ellipsoid segments have been replaced by a piecewise linear representation, to give the muscles a smoother, closed surface. As in [Kähler et al. 2001], skin vertices attach to muscles, and muscles in turn can be attached to the mandible. Since we derive all head models by deformation of the same reference head, we keep those assignments fixed. This makes expressions more reproducible between different head models, because instabilities in vertex re-attachment on the deformed models are avoided. To incorporate elastic properties of muscle fibers, the interaction of merged muscles is modeled more faithfully by using another simple spring mesh. The springs run along the middle axis of each muscle, connecting merged muscles by a common node. Finally, the *orbicularis oris* is divided into two muscles for upper and lower lip, and the center of contraction for each of these can be translated along a line. These added degrees of freedom allow more accurate protrusion and retraction of the lips, which is for example useful in speech animation.

The model is tagged with landmarks, defined on the skin and skull surfaces. We use these landmarks to control deformation of the head structure in our algorithms. The landmarks follow the conventions laid out in [Farkas 1994], where we have chosen a minimum subset of landmarks according to their prominence in the face and existence of a correspondence between skin and skull. There are in general far less landmarks on the skull than on the skin, since not every feature on the skin surface corresponds to one on the skull, cf. Figure 1 d). In our current model, we use 60 skin landmarks and 22 skull landmarks.

4 Animation and rendering overview

The focus of this paper is on the structured head model described in the previous section, and geometric methods of deformation of this model. To establish the context, we nonetheless need to describe the embedding of the model into the complete physics-based facial animation system. This section gives a necessarily brief overview of the issues involved, omitting most of the technical detail.

4.1 Animation control

Facial motion is controlled mainly by specifying muscle contractions over time. We explicitly specify these parameters for a number of keyframes, assembling facial expressions. For animation, we perform interpolation between these contraction values. The complex dynamic of the human face in motion is hard to reproduce in this manner, requiring higher level animation facilities that are beyond the scope of this paper. We have successfully used the muscle-based approach for the automatic generation of speech animation [Albrecht et al. 2002].

4.2 Physics-based simulation

The animation is controlled on the lowest level by muscle parameters from an animation script or from user interaction. During the simulation, the equations of motion for the mass-spring system are numerically integrated through time using a Verlet leapfrog integration scheme [Vesely 1994; Turner and Gobetti 1998]. This explicit forward integration scheme provides better stability than the popular Euler method [Waters and Frisbie 1995; Lee et al. 1995] with similar ease of implementation.

The spring mesh structure is similar to [Kähler et al. 2001], which is advantageous for real-time animation: the complexity is relatively low compared to other layered approaches [Terzopoulos and Waters 1990; Lee et al. 1995]. Volume preservation and skull

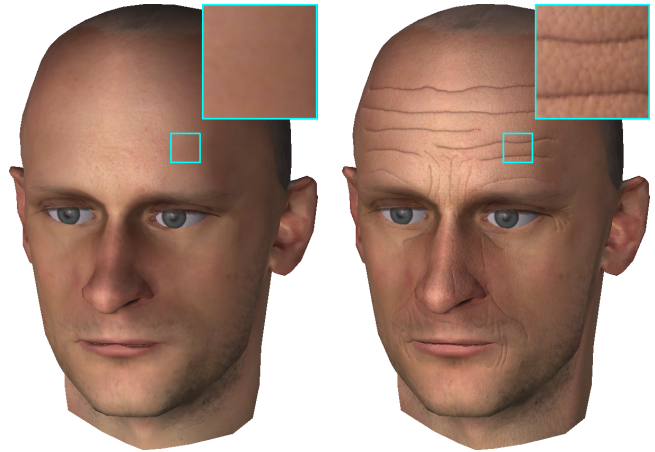


Figure 2: Comparison of an individual with simulated age wrinkles using plain OpenGL rendering (left) and our real-time skin shading algorithm (right).

non-penetration constraints are integrated into the global solution of the equations of motion, obviating the need for local geometric criteria. Since the number of springs in the system is proportional to the number of edges and vertices of the head model, we chose to have a rather low resolution mesh to enable fast simulation updates.

4.3 Multi-threaded simulation and rendering

An important application in our system is speech animation, which requires very fast simulation updates in the range of 40 fps for real-time animation. While we can achieve these rates with our current model, the graphics performance of modern hardware allows for much higher rendering frame rates. Our facial animation system thus decouples simulation and rendering, exploiting dual processor systems by using individual threads for the physics-based simulation and the rendering of an animation [Haber et al. 2001].

4.4 Skin shading

To improve the visual appearance of the skin surface in real-time rendering, we employ the vertex program and register combiner features of the NVidia GeForce3 graphics board. In particular, we apply multitexturing using four texture units. The texture units are assigned to the skin color decal texture, a bump map for the skin surface structure, a bump map for expressive wrinkles, and a *mask texture* that contains different monochrome masks such as a gloss map or a bump intensity map in its color channels. The bump map for the skin structure is computed directly from a synthetic human skin model similar to the one presented in [Ishii et al. 1993]. The wrinkles bump map is automatically created from the layout of the expressive wrinkles, taken from the skin texture. Hardware bump mapping for skin structure and wrinkles is implemented using the OpenGL `NV_vertex_program` and `NV_register_combiners` extensions. In addition, a gloss map is applied to specify the locally varying specular coefficient of a Blinn-Phong shading model, while the intensity of skin dimples over the face is controlled by a bump intensity map. Similar to a gloss map, the latter contains a scalar value per texel to specify the degree to which the bump mapped normal should affect the lighting computation. The whole process is carried out in a single rendering pass on the GeForce3, resulting in frame rates of about 100 fps. Figure 2 shows a comparison of a head model rendered with plain OpenGL capabilities and with our skin shading algorithm.

5 Landmark-based head deformation

Given the head model as described in the Section 3, we are using the landmark information to specify a deformation of the object space so that we can warp the complete head structure to a prescribed target landmark configuration. The details of this deformation are described in this section. In Sections 6 and 7 we demonstrate how this general method can be used for both creation and modification of an animatable head model.

5.1 Setting up the warp function

For deformation of biological tissues, BOOKSTEIN advocates an approach based on thin-plate splines, which minimizes the bending energy of a deformed surface [Bookstein 1997a]. The mechanism can be easily translated to the three-dimensional setting [Bookstein 1997b]. The theory is covered extensively in the literature, so we restrict to the practical construction of the deformation function.

The problem can be stated as one of interpolation: let $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{q}_i \in \mathbb{R}^3$, $i = 1, \dots, n$, be two sets of n landmarks. The *source landmarks* \mathbf{p}_i lie on the geometry we want to deform, and the *target landmarks* \mathbf{q}_i correspond to the features on the target head. We need to find a function \mathbf{f} that maps the \mathbf{p}_i to the \mathbf{q}_i :

$$\mathbf{q}_i = \mathbf{f}(\mathbf{p}_i), \quad i = 1, \dots, n,$$

and which is defined on the volume spanned by the landmarks, so that the function can be used to deform all elements of the head structure. Such a mapping can be expressed by a radial basis function, i.e. a weighted linear combination of n basic functions ϕ_i defined by the source landmark points and an additional explicit affine transformation:

$$\mathbf{f}(\mathbf{p}) = \sum_{i=1}^n \mathbf{c}_i \phi_i(\mathbf{p}) + \mathbf{R}\mathbf{p} + \mathbf{t}, \quad (1)$$

where $\mathbf{p} \in \mathbb{R}^3$ is a point in the volume, $\mathbf{c}_i \in \mathbb{R}^3$ are (unknown) weights, $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ adds rotation, skew, and scaling, and $\mathbf{t} \in \mathbb{R}^3$ is a translation component. Following BOOKSTEIN, we simply use the biharmonic basic function $\phi_i(\mathbf{p}) := \|\mathbf{p} - \mathbf{p}_i\|_2^2$, which minimizes bending energy for the deformation [Duchon 1977]. We have the additional constraints

$$\sum_{i=1}^n \mathbf{c}_i = \mathbf{0} \quad \text{and} \quad \sum_{i=1}^n \mathbf{c}_i^T \mathbf{p}_i = \mathbf{0}$$

to remove affine contributions from the weighted sum of the basic functions [Pighin et al. 1998; Carr et al. 2001].

Setting up a system of linear equations relating source and target landmarks, the unknowns \mathbf{R} , \mathbf{t} , and \mathbf{c}_i can be solved for simultaneously. We first construct three matrices:

$$\mathbf{B} = (\mathbf{q}_1 \quad \dots \quad \mathbf{q}_n \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0})^T \in \mathbb{R}^{(n+4) \times 3},$$

$$\mathbf{P} = \begin{pmatrix} \phi_1(\mathbf{p}_1) & \dots & \phi_n(\mathbf{p}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{p}_n) & \dots & \phi_n(\mathbf{p}_n) \end{pmatrix} \in \mathbb{R}^{n \times n},$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{p}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{p}_n^T & 1 \end{pmatrix} \in \mathbb{R}^{n \times 4}.$$

Now we set up a linear equation system of the form $\mathbf{A}\mathbf{X} = \mathbf{B}$ with

$$\mathbf{A} = \begin{pmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{Q}^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+4) \times (n+4)},$$

$$\mathbf{X} = (\mathbf{c}_1 \quad \dots \quad \mathbf{c}_n \quad \mathbf{R} \quad \mathbf{t})^T \in \mathbb{R}^{(n+4) \times 3}.$$

This linear system is solved using a standard LU decomposition with pivoting. We can now transform a point $\mathbf{p} \in \mathbb{R}^3$ according to Eq. (1).

5.2 Deforming the head structure

Given a warp function defined by landmarks placed on the skin of the source and target heads, we apply this function in different ways to the individual components of the model.

1. The *skin mesh* is deformed by direct application of the function to the vertices of the mesh.
2. The landmarks on the *skull mesh* are related to their counterparts on the skin by a vector, giving an offset from each skull landmark to the corresponding skin landmark, cf. Figure 1 d). When the skin geometry has been fixed, adjusting the local skin thickness thus amounts to changing the scale for such a vector, and deforming the skull mesh accordingly. The deformation function is obtained by offsetting the target skin landmarks along the negated new vectors, resulting in the desired new skull landmark positions. The warp from the current skull landmark positions to these new positions is then applied to the vertices of the skull mesh.
3. *Muscles* in our system are specified by a grid which is initially “painted” onto the skin. The actual shape of the muscles is computed automatically from the available space underneath the skin and follows the geometry of the skin surface. To transfer the muscles to the new geometry, we apply the deformation function to the grid vertices and re-compute the shape. The rebuild process also allows us to accommodate for changes in skin thickness.
4. For the other facial components eyes, teeth, and tongue, we only update position and scale automatically, due to their representation as rigid pieces of geometry in our system. Some fine-tuning is thus necessary to fit them exactly into the deformed model. In principle, if the components are also represented as meshes, the deformation can be applied to their vertices, making this manual step unnecessary.

6 Creating head models from range scans

A primary task in facial modeling is the reproduction of the heads of real individuals. One way to acquire the geometry of a head is to use a range scanning device. In practice, though, it turns out that there are a number of obstacles to using this geometry directly for an animatable model:

- the range data is often noisy and incomplete, especially for structured light scanners, due to projector/camera shadowing effects or bad reflective properties of the surface.
- the geometry is heavily oversampled: direct conversion to a triangle mesh regularly yields hundreds of thousands of polygons. For real-time animation, we need to reduce the complexity to about 3k polygons. Available mesh simplification techniques [Cignoni et al. 1998] unfortunately don’t give enough control over the mesh connectivity to guarantee satisfyingly animatable models. Edges should be properly aligned to facial features and the mesh structure should reflect the basic symmetry of the face.
- some parts relevant for animation cannot be scanned, such as the inner part of the lips.

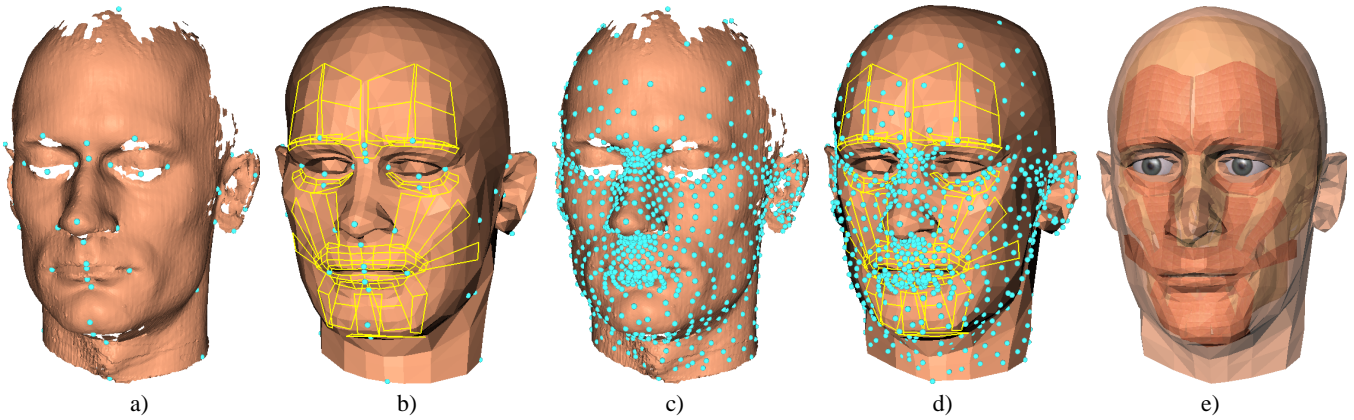


Figure 3: Adaptation of the reference mesh to scan data using feature mesh refinement: a) initial defective target mesh from range scans with landmarks added; b) source mesh and muscle outlines after first deformation; c) target geometry after three refinements; d) deformed source mesh after three refinements and warps; e) final mesh geometry with adapted skull and muscles.

For these reasons, a common solution is the use of a generic face mesh, which is fitted to the target geometry, cf. Section 2. We employ our landmark-based deformation approach to create an animatable model from the reference geometry. No parameterization of the scan data or the reference head mesh is needed, thus we are not restricted to scans from cylindrical range scanners [Lee et al. 1995], but can directly process data from arbitrary sources, and, concerning this, have no restrictions on the topology of the meshes. As a side effect of the procedure, the resulting target model is also tagged with a complete set of anthropometric landmarks, which can be directly used for further deformations, as will be demonstrated in Section 7.

The scan data is given in the form of a dense triangle mesh with no further fixing, such as hole filling, applied. We call this mesh the *target geometry* \mathcal{M}^* in the following, while the reference head (including the structural components) is referred to as the *source geometry* \mathcal{M} — the source geometry needs to be deformed to conform to the target. We proceed as follows:

1. \mathcal{M}^* is tagged with a set of landmarks \mathcal{L}^* corresponding to the set \mathcal{L} defined on \mathcal{M} . This is a “computer-aided” interactive procedure.
2. a deformation is computed based on the landmark correspondences, and \mathcal{M} is warped accordingly.
3. \mathcal{L} and \mathcal{L}^* are automatically refined to generate more correspondences.
4. The components of the reference head model, i.e. skull and muscles, are deformed to match \mathcal{M}^* using the same deformation.
5. Repeat from Step 2 until convergence.
6. muscle shapes are rebuilt and the skull is warped once more to finally adjust the skull/skin relationship, as described in Section 5.2.

In Step 4 of the refinement loop, the skull is deformed using the skin mesh deformation to keep the relation between skin and skull within the accuracy determined by the current density of the landmark sets \mathcal{L} and \mathcal{L}^* . Only in Step 6, the new skull/skin distance is asserted at the sparse locations where landmarks on skin and skull are paired.

We discuss the specification of the landmarks \mathcal{L}^* and the landmark set refinement procedures in detail in the next two sections.

6.1 Specifying landmarks

Our method only requires the specification of a sparse set of landmarks on the target geometry \mathcal{M}^* . Due to the automatic refinement, face features do not need to be laboriously traced using dense correspondences or feature lines [Pighin et al. 1998]. The landmarks are taken from a standard set of the anthropometric literature and are thus well-defined and easy to identify, see Section 3. To ease the “repeated point-and-click” task, we again make use of the deformation function: given three or more landmarks specified manually, we can already set up a mapping from the source set of landmarks \mathcal{L} to the target set \mathcal{L}^* . We then copy and warp all landmarks from \mathcal{L} using this function, resulting in a rough approximation of the desired landmark distribution. Through manual inspection and correction more landmarks are repositioned and fixed in their target positions. The process can be iterated until all landmarks in \mathcal{L}^* have assumed their intended positions. This simple method has shown to be particularly helpful in cases where the scan data is lacking a lot of shape information, since the copied landmarks will already be positioned in a meaningful way. Specifying our reference set of 60 landmarks thus takes only 10–20 minutes in practice. Figure 3 a) shows the scan tagged with the complete set of landmarks, cf. also Figure 1. If no further deformation using this specific set of landmarks is desired, the adaptation of the reference head to the scan data can be performed on an arbitrary set of landmarks.

To further automate the task, we are also actively experimenting with automatic mesh fitting methods, but we feel that there is currently no reliable way to automatically detect the features on the geometry with the required degree of accuracy, especially given the type of incomplete and noisy scan data we have: landmarks often must be placed manually in “mid-air” because the local geometry was not captured by the scanner. Simple heuristics [Noh and Neumann 2001] rely on well-behaved mesh geometry, and are not suitable for most of the anthropometric standard landmarks.

6.2 Adapting the generic mesh

After the initial deformation based on the user-specified landmarks, \mathcal{M} and \mathcal{M}^* are already in good correspondence, see Figure 3 a) and b). But, since the landmark distribution is very sparse, the details in the facial geometry of \mathcal{M}^* are usually not well captured. We do not want to burden the user with specification of hundreds of feature points, so we have developed an automatic procedure that refines the landmark sets \mathcal{L} and \mathcal{L}^* and achieves as good a match as

```

 $\mathcal{M}$   $\leftarrow$  reference head mesh
 $\mathcal{M}^*$   $\leftarrow$  target head mesh
 $\mathcal{L}$   $\leftarrow$  landmark set on reference head
 $\mathcal{L}^*$   $\leftarrow$  landmark set on target head

 $\mathcal{M}$   $\leftarrow$  warp( $\mathcal{L}, \mathcal{L}^*, \mathcal{M}$ ) // deformation from  $\mathcal{L}$  to  $\mathcal{L}^*$  applied to  $\mathcal{M}$ 

 $\mathcal{F}$   $\leftarrow$  feature_mesh( $\mathcal{L}$ ) // construct feature meshes
 $\mathcal{F}^*$   $\leftarrow$  feature_mesh( $\mathcal{L}^*$ ) // using landmark positions

repeat
  ( $\mathcal{F}^*, B$ )  $\leftarrow$  subdivide( $\mathcal{F}^*$ ) // subdivide  $\mathcal{F}^*$ , store in  $B$  baryc. coords
  // of new vertices w.r.t. parent triangles
  ( $\mathcal{F}^*, D$ )  $\leftarrow$  project( $\mathcal{F}^*, \mathcal{M}^*$ ) // project feature vertices onto surface of
  //  $\mathcal{M}^*$  and store displacements in  $D$ 

   $\mathcal{L}^*$   $\leftarrow$  add_landmarks( $\mathcal{F}^*, \mathcal{L}^*$ ) // more target landmarks for
  // appropriate new vertices in  $\mathcal{F}^*$ 

   $\mathcal{F}$   $\leftarrow$  subdiv_copy( $\mathcal{F}, B, D$ ) // subdivide  $\mathcal{F}$  using  $B$  and  $D$ 

  ( $\mathcal{F}, \mathcal{L}$ )  $\leftarrow$  project( $\mathcal{F}, \mathcal{M}$ ) // project feature vertices, landmarks onto  $\mathcal{M}$ 

  flip_edges( $\mathcal{F}$ ) // improve feature mesh smoothness
  flip_edges( $\mathcal{F}^*$ )

   $\mathcal{M}$   $\leftarrow$  warp( $\mathcal{L}, \mathcal{L}^*, \mathcal{M}$ ) // warp using new landmarks
until convergence

```

Table 1: The refinement algorithm for landmark sets on source and target geometries; see Section 6.2 for detailed explanation.

possible in typically two or three iterations.¹ The algorithm outline is shown in Table 1. To be able to refine our standard set of landmarks automatically, we interpret the landmarks as the vertices of a triangle mesh, which we will call a *feature mesh* in the following. Figure 4 shows the layout of this mesh for the geometry of the reference head. One feature mesh \mathcal{F} is constructed for \mathcal{M} , and another one \mathcal{F}^* for \mathcal{M}^* , using the current landmark positions (so they are in fact identical after the first deformation). \mathcal{F}^* is now refined by uniform subdivision: one vertex is inserted into each triangle, splitting it into three new triangles. This vertex is moved to the surface of \mathcal{M}^* , where we take care to find a correct sampling position on the surface, especially in areas of poor scan quality. Often, there is no part of the target surface in the vicinity of the new vertex. If there is, a new landmark is created at the vertex position, and added to \mathcal{L}^* . For each subdivided triangle, the refinement is encoded as the barycentric coordinate of the projection of the new vertex onto the parent triangle along the triangle normal. These coordinates are stored in a set B , and the corresponding scalar displacements along the normal vector in another set D . The right half of Figure 5 outlines the subdivision and projection step for one triangle of \mathcal{M}^* .

We now need to find counterparts on \mathcal{M} for the newly created landmarks. Since \mathcal{F} and \mathcal{F}^* have the same structure, and the geometries are already in good alignment, we repeat the same refinement on \mathcal{F} , using the information from B and D . Each new vertex in \mathcal{F} is now close to the source geometry, but usually not placed exactly on the surface, due to differences in facial detail. If there is a landmark for this vertex in \mathcal{L}^* , we find the nearest intersection along a ray starting at the vertex, correct its position and create a landmark at that point, adding it to \mathcal{L} . The left half of Figure 5 shows how this step is applied.

After all triangles have been refined in this manner, all edges in \mathcal{F} and \mathcal{F}^* from the previous generation are flipped, to improve the

¹“As good as possible” here refers to the limits imposed by the discretization of the source mesh, which is usually much coarser than the scan data.

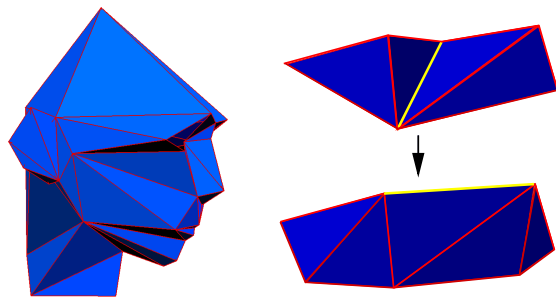


Figure 4: Left: a *feature mesh* is constructed by connecting the landmarks on the head geometry, forming a triangle mesh. Right: flipping edges after subdividing the mesh improves surface smoothness: new vertices on two neighboring subdivided triangles are connected by the yellow edge that previously divided the peaks.

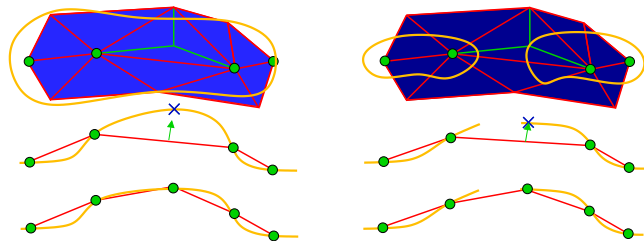


Figure 5: Refining corresponding triangles in the source (left, light blue) and target (right, dark blue) feature meshes. Top: the mesh is typically well-behaved in the source geometry, and fragmented in the target (orange curves). Where vertices of the target feature mesh project onto the geometry, landmarks have been added to both feature meshes (green dots). Target and source feature mesh triangles are refined equally (green edges). Middle: the normal displacement of the target mesh intersection is used to obtain a starting point for finding an intersection with the source mesh. Bottom: source geometry and feature meshes have been deformed to match the landmarks to the target.

quality of the feature mesh surfaces, see Figure 4. The algorithm does not rely on the smoothness of the feature meshes, but if the change in surface normals between adjacent triangles can be kept small during refinement, this will improve the locality of the sampling of the surface. We also filter the triangle normals once by averaging with their neighbors.

Using the two refined versions of \mathcal{L} and \mathcal{L}^* , we set up a new deformation function. Applying it to the source model, and to the corresponding feature mesh, results in a better approximation of the target geometry. The procedure is repeated, and the deformed mesh quickly stabilizes to a good fit of the target geometry. In practice, after only three iterations of the refinement procedure the geometry will have adapted optimally.

Since we use a triangle mesh generated directly from range scan data as the target geometry, we usually have to deal with large areas of the head geometry where there is no data, often interspersed with small specks of samples (e.g. the back of the head), see Figure 3 a). The refinement algorithm is thus geared towards finding these small “islands of data”, while being very conservative in accepting a sampling site on the geometry as a new landmark position. A wrongly placed landmark can cause large distortions in the deformed geometry, rendering it unusable, so we employ heuristics based on surface normals and landmark/surface distance to find and rank the acceptable sites, or reject creation of a landmark. The ray

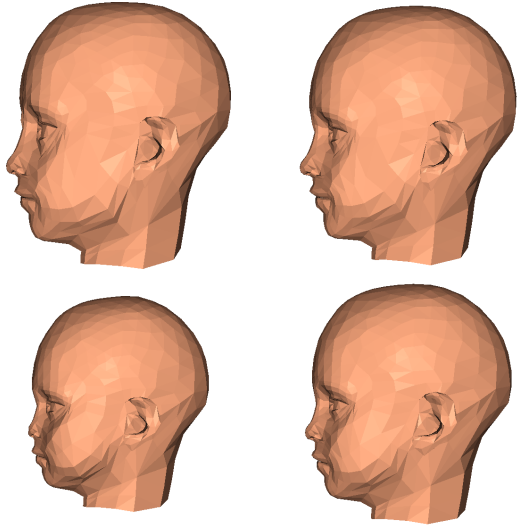


Figure 6: Geometric deformation of a boy’s head by our constraint resolution technique. Clockwise from left top: 20 years, 12 years, 5 years (original age), 1 year.

intersection before repositioning a new vertex in the source feature mesh is much less critical, since we are operating on the deformed reference mesh, which has perfectly well-behaved geometry. Here, we just have to make sure not to intersect with backfacing parts of the geometry. Figure 3 shows the approximation of scanned head geometry by deformation of the reference head.

7 Growth and aging

A challenging problem in facial modeling and animation is the simulation of growth and aging. Besides the Arts, important applications exist in the forensic sciences and medicine: how does the child that got missing ten years ago look now? What are the long-term effects of a rhinoplastic operation? Often, a skilled artist is needed to draw conclusions towards age-related changes from photographs with the help of anthropometric data. Drastic changes in head size and facial proportions occur between childhood and maturity, as well as in skin texture and elasticity of the skin, not to mention hair growth. All of this affects the look of the face in its diverse static poses as well as in motion.

We demonstrate the application of growth and age transformations to the geometry of our animated head model using the deformation technique described in the previous sections. The set of landmarks on the model is used to obtain a variety of standard anthropometric measurements on the head, which are updated according to a user-specified change in age by a constraint-resolution mechanism. Our approach is inspired by DECARLO *et al.* [DeCarlo *et al.* 1998], but has no restrictions on the smoothness or parameterization of the surface. In fact, the computation of the age deformation uses only the landmark set, thus being independent from the deformed surface. In our system, we apply this deformation to the vertices of the head model’s triangle mesh. Also, we do not operate on proportions, but only on distance measurements.

7.1 Landmark measurements

The landmarks on the head model correspond to those in the anthropometric literature used for head measurements. Specifically, we use tabulated measurements for a sample of the North Ameri-

can population of the Caucasian type, males and females between the ages of one year up to twenty-five years (after this age, there is almost no change in facial geometry due to growth) [Farkas 1994]. The data describes distance relations for pairs of landmarks, dependent on age and sex. Three types of distance measurements for a given landmark pair are used in the data:

- distances along one of the horizontal, vertical, and depth directions;
- Euclidean distance;
- arc length, traditionally measured using soft tape on the face.

The head model is placed in the standard posture used for anthropometric measurements, so the axis-aligned distances correspond to the x , y , and z axes in the local coordinate system. Each statistical measurement is given by its mean value μ and standard deviation σ . In our current system, we use 39 axis-aligned distance measurements, 25 Euclidean distances and 6 arc lengths, specified on a part of the standard landmark set on the reference head.

Given age and sex for the current head model, we first compute the current value d_c for a distance measurement directly from the landmarks on the model. The value is compared to the statistical mean value μ_c in the data tables to find its “position” in the assumed standard probability distribution. After looking up the mean value μ_t for the targeted age, we compute the final value d_t at the same relative position in the distribution:

$$d_t = \mu_t + \frac{\sigma_t}{\sigma_c}(d_c - \mu_c),$$

where σ_c and σ_t are standard deviations for the current and target age, respectively. Thus, we retain the characteristics of the source head model even over large changes in age.

For deforming the head geometry, we are now posed with the following problem: given the current landmark positions \mathbf{p}_i and a number of distance measurements, what are the new landmark positions \mathbf{q}_i ($i = 1, \dots, n$)? This problem is largely under-constrained, as there are many solutions that fulfill these *hard constraints*. In our approach, we thus add more constraints between the landmarks that are closest to each other. After deformation, the distances between them should scale roughly with the global scaling s of the head, which we derive from the change in head height. These are *soft constraints*, in that they are used to find the best solution, but they are not strictly enforced.

7.2 Linear constraint resolution

Most of the distance measurements are given along one axis a ($a \in \{x, y, z\}$), which allows us to represent the problem as a set of linear constraints: we want to find the n new landmark coordinates $\mathbf{q}_{a,i}$, $i = 1, \dots, n$, for each axis a . In the following, we derive a solution for one such axis.

The relation to the m hard distance constraints and \tilde{m} soft distance constraints can be expressed by a sparse linear system:

$$\mathbf{A}\mathbf{q}_a = \mathbf{d} + \sum_{i=1}^{\tilde{m}} \lambda_i \tilde{\mathbf{d}}_i,$$

where the combinatorial matrix $\mathbf{A} \in \mathbb{R}^{(m+\tilde{m}) \times n}$ specifies pairings of landmarks. Each row of \mathbf{A} contains exactly two non-zero entries $+1$ and -1 in columns j and k , respectively, to denote the pairing of $\mathbf{q}_{a,j} - \mathbf{q}_{a,k}$. There can be at most $n(n-1)/2$ pairings of landmarks, but in practice, we have $m \approx 35$ plus $\tilde{m} \approx 100$ landmark pairs per axis. The vector $\mathbf{d} \in \mathbb{R}^{m+\tilde{m}}$ represents the hard constraints and has m non-zero entries in those positions, where the

target distance is prescribed. Each vector $\tilde{\mathbf{d}}_i \in \mathbb{R}^{m+\tilde{m}}$ contains a single non-zero entry with the current distance $\mathbf{p}_{a,j} - \mathbf{p}_{a,k}$ between a pair of landmarks not constrained by \mathbf{d} in the corresponding position, i.e. in the row where \mathbf{A} specifies the pairing of $\mathbf{q}_{a,j} - \mathbf{q}_{a,k}$. Since we want to enforce the hard constraints \mathbf{d} given by the data, but keep the soft distances between the other landmarks close to what they were, we solve for weights λ_i close to the global scaling factor s .

The system can be easily reformulated by shifting terms to include the $\tilde{\mathbf{d}}_i$ and λ_i in the matrix:

$$\begin{pmatrix} \mathbf{A} & \tilde{\mathbf{D}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{q}_a \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{s} \end{pmatrix}, \quad (2)$$

where the columns of $\tilde{\mathbf{D}} \in \mathbb{R}^{(m+\tilde{m}) \times \tilde{m}}$ are composed of the vectors $-\tilde{\mathbf{d}}_i$, and $\boldsymbol{\lambda}$ is a vector built from the λ_i in the same order. The submatrix $\mathbf{I} \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$ is an identity matrix. On the right hand side, \mathbf{s} has \tilde{m} entries with the constant scaling factor s . The system is now overconstrained and we solve for the $\mathbf{q}_{a,i}$ and λ_i using a singular value decomposition (SVD) [Press et al. 1992]. Clamping the singular values to achieve a prescribed condition number for the least squares problem before back-substitution allows us to get rid of linearly dependent constraints, as they occur in practice.

According to this method, we set up and solve three independent linear systems for the distance constraints along the x , y , and z axes. Since the data is only as exact as the procedures used for taking measurements of the sample population, and the collected data is only statistical, a precise solution for a given individual head can not in general be achieved. However, SVD will give a best fit in the least squares sense: for a system $\mathbf{Ax} = \mathbf{b}$, the solution vector \mathbf{x} that minimizes the residual error $\|\mathbf{Ax} - \mathbf{b}\|_2$ will be found. In Eq. (2), the values in \mathbf{d} are typically in the range of 10–200 mm, while the values in \mathbf{s} are close to 1.0. Thus, a small error (i.e. a displacement from the ‘ideal’ target position) in one of the new landmark coordinates $\mathbf{q}_{a,i}$ results in a much larger residual error than a small deviation in one of the weights λ_i . As a result, we found the hard constraints to be fulfilled with a maximum absolute error of two millimeters in our experiments.

7.3 Non-linear constraints

For some landmark pairs, a Euclidean distance is given, which can not be included directly into the linear system. We linearize the problem by splitting such a constraint into three axis-aligned constraints. Given a current vector from one landmark to another and a prescribed target distance between them, we assume that the direction of that vector will not change drastically in the solution. We scale this vector to the target length and project it onto the three axes of the global coordinate system. We add the three projected distances as additional linear constraints into the equations described in the previous section and let SVD run as before, arriving at a solution that approximately fulfills the Euclidean distance constraint. To improve the solution, we repeat the projection process and solve again, until convergence. In practice, three iterations suffice.

Arc lengths are another measurement in the data. Since the pairs connected by arcs are constrained additionally by distance measurements in the table data, we do not include the arc lengths in the constraint resolution mechanism. Instead, we use the arc measurements only to improve the shape of the surface *after* solving. The arc is approximated by a circle segment: we use a virtual ‘‘middle landmark’’ that is placed between the two landmarks connected by the arc. This landmark is shifted along the surface normal, to give the circle segment the arc length specified in the data.

8 Results and conclusion

We have presented a head model with anatomical structure and deformation methods that allow adaptation of such a model to individual scan data, and simulation of human head growth. Obviously, without medical volumetric data, the finer individual details of the head anatomy cannot be guessed by our method. Still, without such expensive medical hardware as a CT scanner, we are able to produce a more plausible reconstruction of these internal structures than previous approaches that estimate the skull as an offset surface from the skin or use simpler models of muscle geometry.

The landmark-based fitting has shown to work robustly on raw meshes obtained from range scans of a variety of individuals. The interactive specification of the initial sparse landmark set has shown to be advantageous for creating models from incomplete data, where large parts of the geometry cannot be estimated automatically in a reliable way. The robust adaptation to this kind of input data makes mesh fixing and hole filling abdicable, a process that can otherwise easily take several hours of interactive work.

The computational cost of the landmark refinement algorithm largely depends on ray/mesh intersection tests and point/triangle distance computations. We expect a great speed-up from optimization of these tests. In our current implementation, we arrive at about five minutes total run time for the fitting process on a 1 GHz PC with a 100k triangle target mesh. Given the scan data, the whole process of creating an animatable head model including the tuning of eye, teeth, and tongue positions takes 20–30 minutes in our experience.

Further deformation of the face for changing the age produces plausible results, which is encouraging given that only a small amount of statistical data is used. Our method assumes that a measurement keeps its variance from the statistical mean over the years: a nose that is relatively big for an adult, is assumed to be big in childhood. Together with the scarcity of the facial measurements, this tends to retain the characteristics of the original face to a sometimes too strong degree. Also, examination of the tabulated arc measurements delivered surprising results: the ratio between arc length and corresponding Euclidean distance remains almost constant through the ages in the table data, i.e. the roundness of the face does not vary significantly, different from what one would expect especially for younger faces. To incorporate the ‘‘puffiness’’ of small childrens’ cheeks that can be observed in the real world, we allowed for an adjustable slight increase in the arc length up to 10% over the original values for a young child.

Transfer of expressions and animations between generated models using the same initial muscle set has shown to work well even over widely varying age ranges. The common muscle set and parameterization of all our head models simplify the creation and use of a generic expression library a great deal. After adaptation of the model, this parameterization can be used for further editing, which is an advantage over purely geometric transfer of motion between models [Noh and Neumann 2001]: individual characteristics of facial motion, such as a particular way of smiling, can be included by manipulation of the same parameters, instead of working on the vertex data of the new model. Also, we see some automation potential for dealing with age-related characteristics based on the muscle parameterization: e.g. for a young child, we globally scale down facial muscle movements, to accommodate for the not yet fully developed abilities of expression.

9 Future work

Skin stiffness constants, skin thickness and muscle layer thickness are evaluated and used for each deformation of the reference head to re-shape the skull and the muscle layer, and to initialize the spring mesh. We currently have to adjust these parameters manually to

accommodate for individual and age-related changes. Given more statistical data, it should be possible to have these values computed automatically. Also, the visual impact of such adjustments onto the actual animation needs to be examined.

For further automation and enhancement of the precision of the landmark specification process, we would like to make use of the information contained in the photographs used for texturing. It also is appealing to use the landmarks on the fitted face model for this texturing step, which is currently a separate procedure. This would also apply to the synthesis of wrinkles and skin structure. We are investigating if image-based techniques such as proposed by TIDDEMAN *et al.* [Tiddeman *et al.* 2001] can be used to minimize manual texture processing for wrinkle definition.

Currently, we cannot predict the skin color of an individual at an age different from the age of data acquisition. Some heuristics could be applied: babies of Caucasian type typically have a pale, pinkish skin color, while adults often have suntanned skin. Some more reliable results could probably be obtained by applying age-dependent skin parameters such as moisture or elasticity to a micro geometry skin model and using a BRDF-based rendering approach.

Our results indicate that “anthropometric modeling” is a fruitful approach, and could become a useful tool for artists and scientists alike. More detailed and precise data, and a better understanding of age-related changes contained in this data are needed, though. Based on our constraint resolution technique, new face modeling tools can be devised that allow specification of facial feature relations either directly, or indirectly by age, gender, or other statistically captured variables.

References

- ALBRECHT, I., HABER, J., AND SEIDEL, H.-P. 2002. Speech Synchronization for Physics-based Facial Animation. In *Proc. WSCG 2002*, 9–16.
- BLANZ, V., AND VETTER, T. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Computer Graphics (SIGGRAPH '99 Conf. Proc.)*, 187–194.
- BOOKSTEIN, F. L. 1997. *Morphometric Tools for Landmark Data*. Cambridge University Press.
- BOOKSTEIN, F. L. 1997. Shape and the Information in Medical Images: A Decade of the Morphometric Synthesis. *Computer Vision and Image Understanding* 66, 2, 97–118.
- CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and Representation of 3D Objects With Radial Basis Functions. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, ACM SIGGRAPH, 67–76.
- CHADWICK, J. E., HAUMANN, D. R., AND PARENT, R. E. 1989. Layered Construction for Deformable Animated Characters. In *Computer Graphics (SIGGRAPH '89 Conf. Proc.)*, vol. 23, 243–252.
- CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1998. A comparison of mesh simplification algorithms. *Computers & Graphics* 22, 1, 37–54.
- DECARLO, D., METAXAS, D., AND STONE, M. 1998. An Anthropometric Face Model using Variational Techniques. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, 67–74.
- DUCHON, J. 1977. Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller, Eds., vol. 571 of *Lecture Notes in Mathematics*, 85–100.
- FARKAS, L. G. 1994. *Anthropometry of the Head and Face*, 2nd ed. Raven Press.
- HABER, J., KÄHLER, K., ALBRECHT, I., YAMAUCHI, H., AND SEIDEL, H.-P. 2001. Face to Face: From Real Humans to Realistic Facial Animation. In *Proc. Israel-Korea Binational Conference on Geometrical Modeling and Computer Graphics*, 73–82.
- ISHII, T., YASUDA, T., YOKOI, S., AND TORIWAKI, J. 1993. A Generation Model for Human Skin Texture. In *Proc. Computer Graphics International '93*, 139–150.
- KÄHLER, K., HABER, J., AND SEIDEL, H.-P. 2001. Geometry-based Muscle Modeling for Facial Animation. In *Proc. Graphics Interface 2001*, 37–46.
- KOCH, R. M., GROSS, M. H., AND BOSSHARD, A. A. 1998. Emotion Editing using Finite Elements. In *Computer Graphics Forum (Proc. Eurographics '98)*, vol. 17, C295–C302.
- LANITIS, A., TAYLOR, C., AND COOTES, T. 1999. Modeling the process of ageing in face images. In *Proc. 7th IEEE International Conference on Computer Vision*, IEEE, vol. I, 131–136.
- LEE, Y., TERZOPOULOS, D., AND WATERS, K. 1993. Constructing Physics-based Facial Models of Individuals. In *Proc. Graphics Interface '93*, 1–8.
- LEE, Y., TERZOPOULOS, D., AND WATERS, K. 1995. Realistic Modeling for Facial Animation. In *Computer Graphics (SIGGRAPH '95 Conf. Proc.)*, 55–62.
- LEE, W.-S., WU, Y., AND MAGNENAT-THALMANN, N. 1999. Cloning and Aging in a VR Family. In *Proc. IEEE Virtual Reality '99*, 13–17.
- NOH, J., AND NEUMANN, U. 2001. Expression cloning. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, ACM SIGGRAPH, 277–288.
- PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI, R., AND SALESIN, D. H. 1998. Synthesizing Realistic Facial Expressions from Photographs. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, 75–84.
- PLATT, S. M., AND BADLER, N. I. 1981. Animating Facial Expressions. In *Computer Graphics (SIGGRAPH '81 Conf. Proc.)*, vol. 15, 245–252.
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2. ed. Cambridge University Press, Cambridge, MA.
- SCHEEPERS, F., PARENT, R. E., CARLSON, W. E., AND MAY, S. F. 1997. Anatomy-Based Modeling of the Human Musculature. In *Computer Graphics (SIGGRAPH '97 Conf. Proc.)*, 163–172.
- SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-Form Deformation of Solid Geometric Models. In *Computer Graphics (SIGGRAPH '86 Conf. Proc.)*, vol. 20, 151–160.
- SZELISKI, R., AND LAVALLÉE, S. 1996. Matching 3-D Anatomical Surfaces with Non-Rigid Deformations using Octree-Splines. *International Journal of Computer Vision* 18, 2, 171–186.
- TERZOPOULOS, D., AND WATERS, K. 1990. Physically-based Facial Modelling, Analysis, and Animation. *Journal of Visualization and Computer Animation* 1, 2 (Dec.), 73–80.
- TIDDEMAN, B., BURT, M., AND PERRETT, D. 2001. Prototyping and Transforming Facial Textures for Perception Research. *IEEE Computer Graphics and Applications* 21, 5 (Sept./Oct.), 42–50.
- TURNER, R., AND GOBBETTI, E. 1998. Interactive construction and animation of layered elastically deformable characters. *Computer Graphics Forum* 17, 2, 135–152.
- VESELY, F. J. 1994. *Computational Physics: An Introduction*. Plenum Press, New York.
- WATERS, K., AND FRISBIE, J. 1995. A Coordinated Muscle Model for Speech Animation. In *Proc. Graphics Interface '95*, 163–170.
- WATERS, K. 1987. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics (SIGGRAPH '87 Conf. Proc.)*, vol. 21, 17–24.
- WILHELMS, J., AND VAN GELDER, A. 1997. Anatomically Based Modeling. In *Computer Graphics (SIGGRAPH '97 Conf. Proc.)*, 173–180.
- WU, Y., MAGNENAT-THALMANN, N., AND THALMANN, D. 1994. A Plastic-Visco-Elastic Model for Wrinkles in Facial Animation and Skin Aging. In *Proc. Pacific Graphics '94*, 201–214.
- WU, Y., KALRA, P., MOCCOZET, L., AND MAGNENAT-THALMANN, N. 1999. Simulating Wrinkles and Skin Aging. *The Visual Computer* 15, 4, 183–198.