
MEDUSA — A Facial Modeling and Animation System

Jörg Haber
Max-Planck-Institut für Informatik, Saarbrücken

Abstract

We present a system for photo-realistic facial modeling and animation, which includes several tools that facilitate necessary tasks such as mesh processing, texture registration, and assembling of facial components. The resulting head model reflects the anatomical structure of the human head including skull, skin, and muscles. Semi-automatic generation of high-quality models from scan data for physics-based animation becomes possible with little effort.

A state-of-the-art speech synchronization technique is integrated into our system, resulting in realistic speech animations that can be rendered at real-time frame rates on current PC hardware.

Keywords: facial animation, physics-based simulation, speech synchronization

1 Introduction

Modeling and animation of human faces is one of the most difficult tasks in computer graphics today, even more so when life is to be breathed into digitized versions of real, well-known individuals. With the advent of virtual social spaces, where people communicate face-to-face, the demand for believable virtual characters increases. Accurate reconstruction of individual faces also has major applications in medicine, e.g. cosmetic surgery, and entertainment, e.g. virtual actors.

Why is this goal so elusive? Part of the answer lies in the sensitivity of the human visual system to the nuances of facial expression that it can perceive. Another part is the sheer size of the task of reproducing the complexity of the face in the computer. An individual’s face needs to be accurately captured in geometry and texture. To animate the face on a low level, appropriate animation parameters have to be chosen and the resulting face deformations must be computed. On a higher level, these animation parameters are used to produce expressions and speech. Finally, the animated face needs to be rendered. Apart from the conceptual complexity of each of these tasks, further constraints arise when real-time animation is required, as it is the case for dynamic virtual environments.

In this paper, we present an overview of our facial animation system MEDUSA, discussing each step from data acquisition to real-time rendering. We aim at the construction of realistic animatable head models from real human individuals. Geometry and images are acquired in high resolution and converted to textured polygonal geometry, simplified according to the requirements of the targeted hardware. On the lowest level, we use a physics-based approach with muscle contraction values as animation parameters. The construction of the virtual head model is based on the human anatomy, and we have developed tools to automate the task of adapting and linking this model to the actual face geometry. Multi-threaded execution of simulation and rendering results in real-time frame rates on current PC hardware. A schematic overview of the animation components of our system is depicted in Figure 1 on page 15. As an application demonstrating higher level animation, we drive the system using a state-of-the-art speech synchronization method that takes into account the complex influence of coarticulation.

2 Previous and Related Work

Animating models of the human face has remained an attractive and challenging area of research over the last 25 years [26, 30, 27]. A multitude of techniques and approaches are documented in the literature, which can be broadly classified into the following categories: parametric models, physics-based models, image-based methods, and performance-based animation.

Parametric models have been invented very early, to avoid the complexity of specifying complete key frames for each facial posture. Deformation of the skin is achieved by direct manipulation of the geometry [26, 27]. The parameterizations are often inspired by anatomical knowledge: vectors and radial functions have been used by WATERS [35] to approximate deformations caused by the facial musculature. Free-form deformations have been employed by CHADWICK *et al.* [5] to shape the skin in a multi-layer model,

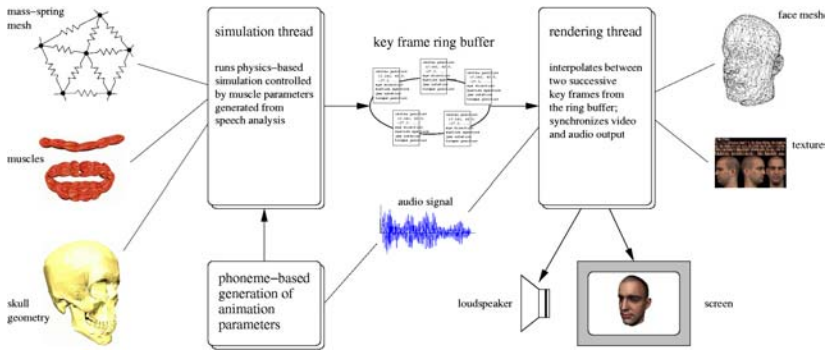


Fig. 1: Overview of the simulation and rendering components of our system. Inter-thread communication between the simulation and the rendering thread is established via the key frame ring buffer, cf. Section 5.3.

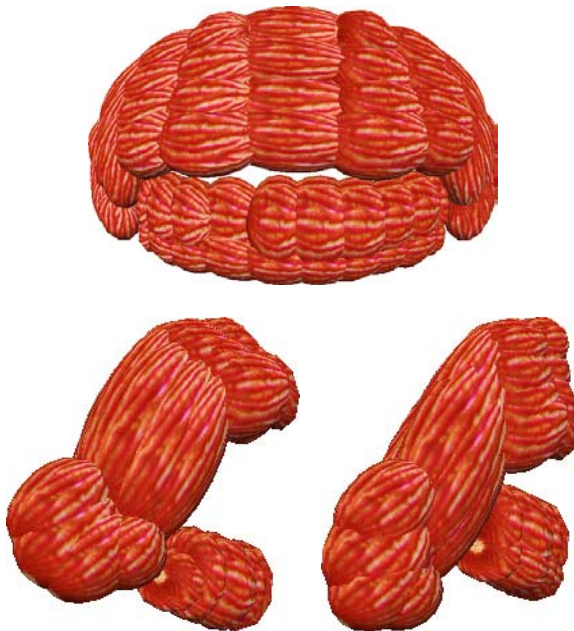


Fig. 2: Two-part orbicularis oris model. Top: relaxed state with closed mouth, front view. Bottom left: protruded lips, slightly opened mouth, side view. Bottom right: upper lip retracted, lower lip moved upward and inward.

which contains bones, muscles, fat tissue, and skin. NAHAS *et al.* [25] use a B-spline surface model to generate synthetic visual speech. A standardized set of facial animation parameters has recently been established in the form of the MPEG-4 standard [15], which has been used by GOTO *et al.* [11] to control their facial animation system.

Building on the idea of virtual muscles, physics-based approaches attempt to simulate 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 [30, 21, 20]. In the context of speech animation, WATERS and FRISBIE [36] proposed a two-dimensional mass-spring model of the mouth with the muscles represented as bands. TERZOPOULOS and WATERS [33] automatically construct a three-dimensional model of the human face from an initial triangle mesh. Their model consists of three layers representing the muscle layer, dermis and epidermis. The elastic properties of skin are simulated using a mass-spring system. Employing additional volume preservation and skull penetration constraints, this approach produces realistic effects including wrinkling at interactive frame rates. This model was later simplified by LEE *et al.* [22] to two layers (dermal-fatty and muscles), which connect to the bone structure underneath. Construction of the head model from acquired surface data is largely automated. WU *et al.* focus on generation of expressive wrinkles and skin aging effects. Their face model incorporates viscoelastic properties of skin, and muscles are represented by surfaces of revolution [40] or B-spline patches [39]. Accurate simulation of skeletal muscles (not regarding the overlying skin tissue) has been developed by CHEN and ZELTZER [6] based on a finite element model. Recently, SCHEEPERS *et al.* [32] and WILHELMS and VAN GELDER [37] introduced anatomy-based muscle models for animating humans and animals, also focusing on the skeletal musculature. Skin tissue is represented only by an implicit surface with zero thickness [37].

The computational complexity and the necessary anatomical knowledge for accurate physics-based simulation of facial expressions have recently led to the rise of image-based techniques [29, 4]. These approaches result in photo-realistic images by matching a three-dimensional model to one or more photographs of an individual. Animation is still limited, though, since each expression has to be captured in advance and blending expressions is achieved by linear blends, neglecting the dynamics of facial gestures. Capturing these dynamics is the essence of performance-based methods [38, 12], where real faces are tracked to reproduce the motion on the virtual model.

Several facial animation systems that include speech synchronization have been proposed. LEWIS and PARKE presented automatic speech synchronization for recorded speech [23]. They use linear prediction to analyze the speech signal and generate a phoneme sequence from a set of twelve reference

phonemes. Each phoneme is associated with parameters for lip shape and jaw rotation, which are used to render key frames of a parametric face model. An automatic approach to synthesize speech by rules and drive a parametric face model has been proposed by PEARCE, HILL *et al.* [28, 13]. Parameter values for the animation were taken from measuring and comparing front view photographs. This approach has been adopted by COHEN and MASSARO [7]. They enhanced the model by modeling coarticulation and using a synthetic tongue. KALRA *et al.* describe a multi-layered approach to specify facial animation [18]. In their model, words are mapped to phonemes using an interactively built dictionary. Phonemes and expressions are in turn translated into abstract muscle action procedures that drive facial animation. A video showing a real person talking was used by WATERS to generate control parameters for his two-dimensional mouth model [36]. He uses an optimization algorithm to determine muscle parameters that minimize the total distance between seven reference points on the digitized frames of the video and the corresponding points on the model. IP and CHAN use an English-text-to-phoneme parser to generate a sequence of phonemes [14]. Using an interactively built database, each phoneme is translated into a set of control point displacements, which are applied to a face model represented by a NURBS surface.

3 Data Acquisition

In our approach, facial data is captured from real humans. Head geometry and texture information are acquired in separate procedures. Geometry acquisition results in a high resolution triangle mesh, against which textures are registered. For real-time applications, the geometry is scaled down using mesh simplification to any desired level, and the final textured model is obtained by automatically re-registering the textures to the geometry. In this way, we can quickly create a number of models with different resolutions in terms of the polygon count.

3.1 Geometry

Data is acquired using a range scanning system based on the triangulation principle. The scanner delivers uncalibrated range images. The data is processed over a couple of steps to obtain the final head geometry:

1. Range images are registered, resulting in a point cloud.
2. An initial triangle mesh is generated by surface extraction.
3. General post-processing is usually necessary to fix holes due to missing data and to remove noise.

4. The geometry needs to be prepared for facial animation: the range scans are taken from a model with a neutral expression and a closed mouth. Thus we have to cut the mesh open between the lips. In addition, the part of the mesh representing the eyeballs is flattened to allow for proper insertion of our synthetic eye models, cf. Section 4.1.
5. Mesh decimation is applied to the once-prepared high resolution mesh as needed, to produce approximations that are both suitable for simulation and real-time rendering purposes. We have obtained good results with about 3000–4000 triangles for a complete head model.

Methods for processing polygonal geometry in this manner are well described in the literature [19]. We give some details here about the mesh decimation algorithm, which has to preserve a number of properties on the mesh, in particular:

Approximation error: the resulting face mesh has to approximate the original geometry closely, not only to serve resemblance to the original, but also to be able to re-apply the registration parameters for the textures. We found a tolerance of 1–2 mm distance error (from the point of the original mesh to the surface of the new mesh) to be practical. This is actually within the range of error present in the data obtained from the range scanner.

Triangle quality: the numerical stability of the animation system is improved by generating triangles with similar edge length. Also, the triangles should not be too long, regardless of approximation quality, so that local vertex movements during animation do not influence far away regions of the mesh.

Feature preservation: critical regions in the human face are the eyes and the mouth. Here, a higher resolution of the mesh is necessary to preserve good quality under deformation. We have found that limiting changes in mesh curvature (based on measuring face normals) are appropriate to keep these parts of the face sufficiently detailed.

Figure 7 a) shows two snapshots of our mesh decimation tool with the initial geometry and a low-resolution approximation of the head model.

3.2 Textures

To generate a *texture*¹ for a head model, we take several photographs of the person’s head using different, uncalibrated camera positions. The number of photographs we use varies from four to eleven, depending on the desired quality of the combined texture and the time available for texture registration. In practice, we found four to six photographs to be sufficient. All photographs are taken with a high resolution digital camera under diffuse illumination.

¹an electronic image that contains color values for the surface to which it is mapped

During the photo session, the facial expression of the person should resemble the neutral expression during the scanning process.

The photographs are registered and combined into a single texture suitable for rendering on graphics hardware (OpenGL) similarly to the approach in [31], which is based on the camera calibration technique developed by TSAI [34]. In contrast to the method proposed in [31], we do not rely on the property that every vertex of a face mesh is bound to an input texture. Especially in the regions inside and behind the ears, some vertices typically remain unbound. For each unbound vertex v in the mesh, we perform a region growing over the topological neighborhood of v and interpolate the texture coordinates of those vertices within that region R_v , which are bound to the same and most frequently used texture. If the texture binding within R_v is evenly distributed among several textures, we examine remaining *valid* texture bindings of the vertices in R_v (cf. [31, Sect. 3] for a definition of *valid*) and interpolate the texture coordinates within the prevailing texture. If there is still no preferred texture among all texture bindings, we randomly choose one of the most frequently used textures. The interpolated texture coordinates of vertex v need to be verified to lie within $[0, 1]^2$. If this is not the case, another frequently used texture is chosen for interpolation. Figure 7 b) shows a snapshot of the GUI of our texture registration tool, where corresponding points have been interactively selected on both the 3D geometry and the 2D input photograph.

4 Construction of the Head Model

4.1 Eyes, Teeth, and Tongue

What do the human eyes, teeth, and tongue have in common? They are both important for realistic facial animation, and it is difficult to acquire data from a human being to precisely model these facial components. Thus we use generic models of these components, see Figure 8. The design of our generic models has been chosen such that they look convincingly realistic when inserted into a face mesh while still being rendered efficiently using OpenGL hardware. We have acquired a set of textures to choose from, e.g. blue, green, and brown eyes. Currently, we are working on the development of a texture synthesis algorithm, which generates a full eye texture from the (small) part of the eye that is visible in a standard portrait photograph.

Our generic facial components are animatable in a number of ways. For the eyes, the viewing direction can be specified as well as the size of the pupil and the aperture of the eyelids. The latter is also used to control the brightness of the eyes. While the position of the upper teeth remains fixed with respect to

the skull position, the lower teeth can rotate along with the mandible about the axis that connects the left and right temporomandibular joint. An additional horizontal translation can be specified for the mandible and the lower teeth. Our tongue model allows rigid transformations, which are composed with the transformation of the mandible. We are currently investigating the applicability of more powerful shape-deforming transformations for the tongue. The brightness of both the teeth and the tongue is scaled by the mouth opening value.

4.2 *Muscles*

In our physics-based simulation, deformation of the skin mesh is caused by simulated contraction of facial muscles. For speech animation, we have built a set consisting of twelve of the most important muscles in the lower face. These muscles have been modeled from interactively specified coarse outlines on one of our head models using our muscle modeling approach as described in [17], see also Figure 7 c). Details on the behavior of the muscles during animation are given in Section 5.1.

4.3 *Fitting of Components*

The need to match all the parts described above to a specific head model results in a sizable amount of work. To ease the construction of the head model, we express all components in the coordinate frame of a generic reference skull (see Figure 9).

Upon creation of a new virtual head model, the reference skull is matched interactively to the skin mesh using affine transformations. By applying the skull transformation to the parts², we get an initial layout for all components of the head model. Adaptation of the muscle set to the details of the face geometry and linking the mesh to the muscles proceeds fully automatically. Usually only small corrections are necessary to accommodate for individual features of a face, which we support in our interactive editing tool depicted in Figure 7 c).

5 Animation and Rendering

5.1 *Muscle Animation*

Each muscle is built from individual fibers that are in turn composed of piecewise linear segments, given as a control polygon. An ellipsoid is aligned to

²only teeth and tongue are deformed, eyes are only uniformly scaled

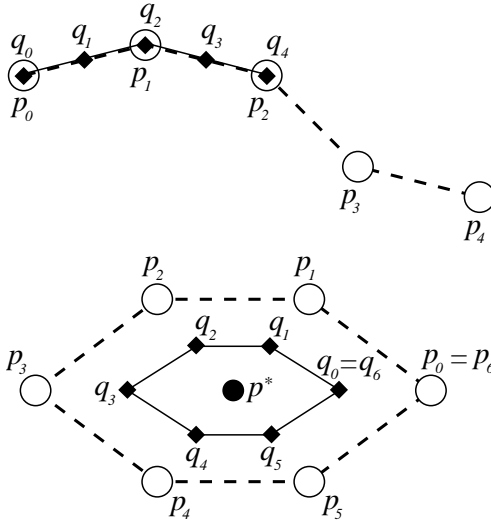


Fig. 3: Contraction ($c = \frac{1}{2}$) of a linear (top) and a sphincter (bottom) muscle fiber. The control points $\{p_i\}$ and $\{q_i\}$ represent the relaxed and contracted muscle, respectively.

each of these segments to provide shape and volume. Given a contraction parameter $c \in [0, 1]$, a new control polygon is computed from the initial one. Sphincter muscles are supported by contraction towards a center point. Through recomputation of the assigned quadrics, where we also incorporate bulging effects, the muscle is re-shaped to reflect its contracted state [17]. Figure 3 exemplifies contraction of a linear and a sphincter muscle fiber.

In speech animation, the most important muscle is the sphincter enclosing the lips, the *orbicularis oris*. This muscle is usually approximated as a closed ring, though from an anatomical point of view it consists of at least four interconnected parts. Many of the parallel muscles from the lower face merge into the orbicularis oris, e.g. the lip raisers (the *zygomaticus* and *levator labii superioris* muscles). In articulated speech, the shape of the mouth is mostly determined by the shape of the surrounding muscles. A number of observations can be made about their deformation:

Elastic behavior: real muscles straighten under tension, i.e. a parallel muscle that has a curved shape in relaxed state resembles a rather straight line under contraction (resulting from the forces applied to both ends of the muscle). Also, interconnected muscles apply forces to each other, which leads to additional deformations: for instance, when the lip raisers contract to form a smile, the upper and lower lip muscle is stretched as well, and the lower lip follows the upward movement.

Bulging: contracted muscles become thicker; this is very visible for instance in the orbicularis oris.

Non-homogeneous contraction: muscles are composed of many fibers, which usually contract by an equal or similar amount. But especially during speech, the orbicularis oris is capable of much more articulated motion resulting from different contraction of the fibers in the various segments. Lips can be protruded or retracted, and upper and lower lip are able to move independently. For instance, during the articulation of an /f/, the upper lip is protruded much more than the lower lip.

These observations suggest a much more complex muscle model based on physical properties such as elasticity. However, in the context of real-time animation, the computational overhead is prohibitive. To produce realistic speech animation, we have improved our earlier muscle model [17] in a number of ways to increase the range of expression especially around the mouth. Few additional computations are necessary, so the model is still very efficient. These modifications are:

Straightening: the control polygon of a linear muscle fiber converges to a straight line under tension. This is achieved by simple linear interpolation based on the original and current distances of the muscle end points: when the Euclidean distance between the endpoints exceeds the length of the control polygon segments in relaxed state, the muscle is completely stretched out along the line connecting the endpoints.

Central axis for sphincters: for reproducing protrusion and retraction of the orbicularis oris, a static central point of contraction is not sufficient. Thus, we specify a central *axis* instead, and muscle deformation is now controlled by two parameters: *contraction* specifies the shrinking orthogonally to the axis, and *protrusion* specifies movement along the axis in either direction, so the lips can even be “tucked in”.

Protrusion gradient: to accommodate for the non-homogeneous contraction of the orbicularis oris, we allow the fibers of a sphincter to protrude by different amounts. The outer part of the sphincter protrudes very little since around the mouth this part is constrained by other muscles and attachments to skin tissue, while the inner part takes on the full protrusion value specified, reflecting the freedom of movement at the lips. Between the extremes we linearly interpolate the protrusion value.

Constraint resolution: our original muscle model already handles merging of muscles in a simple manner: the position of the control points in the merged area of two muscles is a weighted average of the individually computed positions of contracted control points. This local mechanism of making muscles “stick together” has been extended by linking the connection points among each other by simplified springs with a rest length and a stiffness. After resolving constraints locally, the changes to the node positions

induce changes in the lengths of these springs. We now directly compute new point positions from the distortion of the springs and their stiffness values over a few iterations, thus distributing the original displacements among the connected nodes. The number of iterations depends on the maximum distance of nodes in the connection graph. Typically, three or four iterations are sufficient.

In addition to these general changes to our muscle simulation model, we have split the orbicularis oris into two parts for the upper and lower lip. This allows for independent specification of contraction and protrusion values. Figure 2 shows some postures of the orbicularis oris that can be achieved with our improved model.

5.2 *Speech Synchronization*

A crucial point in speech synchronization is the provision for *coarticulation*. This term refers to the influence of the surrounding segments of a phoneme onto the vocal tract shape. For instance, the /k/ in ‘coin’ and the /k/ in ‘cow’ are quite distinct. In ‘coin’, the lip rounding for the /ɔi/ does not begin with the ‘o’ but already with the ‘c’, whereas the /aʊs/ in ‘cow’ does not affect the shape of the lips for the ‘c’. The influencing phonemes may be as many as seven segments apart [2] and may even be separated by syllable or word boundaries [3]. This interlocking of phonemes blurs the acoustic segment boundaries to such an extent that it is impossible to tell exactly where one segment begins and the previous one ends.

Our speech synchronization [1] is based on the approach of COHEN and MASSARO [7], which has also been used recently for the visual speech component of the talking face in the CSLU toolkit [8]. This toolkit was designed to assist teachers of profoundly deaf children with their daily lessons. For example by watching the artificial head pronounce a sentence, the children can improve their own pronunciation.

The lip synch method in [7] requires as input the transcription of an audio file, where the transcription contains both the phonemes and the corresponding timing information. In our system, we obtain the segmentation of the utterances from the ESPS signal processing package [9] and encode the labeling using the TIMIT database notation [10].

Coarticulation is modeled using dominance functions. They describe the influence of a speech segment on the vocal tract shape over time. In the original linguistic theory [24], each segment has a different dominance over every articulator. The approach in [7] uses facial parameters such as, for instance, lip protrusion to model the articulators. In our physics-based muscle model, we use muscle contraction and protrusion parameters for five decisive muscles around the mouth instead of a direct parameterization. In addition, we set

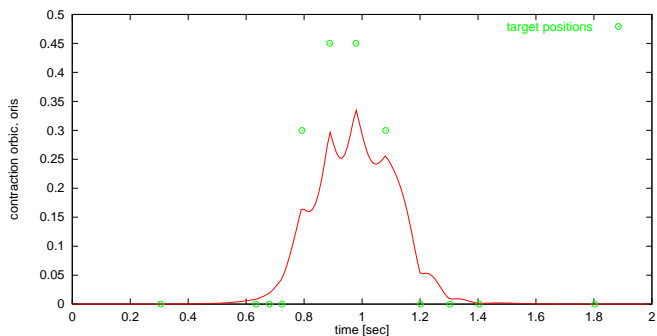


Fig. 4: Resulting function from computing the weighted average of the dominance functions multiplied with the target positions of the contraction of the orbicularis oris for each phoneme of the utterance “hello world”.

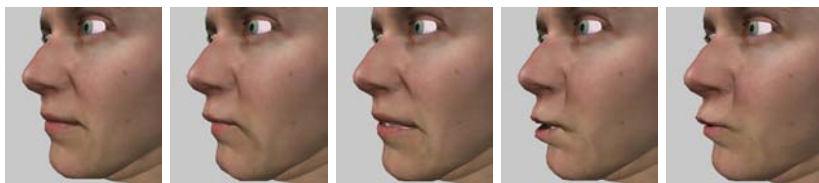


Fig. 5: Snapshots showing mouth articulation. Left to right: neutral pose, /f/, /i/, /b/, /w/. Note how upper and lower lip move independently and are able to not only contract, but protrude or retract.



Fig. 6: Snapshot of the mouth forming a /ð/ as it is pronounced in the English word ‘the’. The tip of the tongue touches the bottom of the upper teeth.

the jaw rotation angle and the transformation parameters of the tongue. Each of these parameters is controlled by an individual function that is computed as a weighted average of the dominance functions over all segments multiplied with the targets of the according muscle (or jaw rotation) for each phoneme. Figure 4 shows such a function for the contraction value of the orbicularis oris for the utterance “hello world”. The target of a muscle for a certain phoneme is given by the contraction and/or protrusion the muscle should take on if the phoneme was pronounced isolatedly. As in natural speech, the targets are hardly ever fully reached. Dominance functions of nearby segments may overlap, leading to an overlapping of the corresponding speech gestures, which in turn leads to coarticulation.

The time available for tightening and relaxation of muscle contraction also determines the amount of the contraction: in fast speech there is more coarticulation than in slow speech. This is reflected by a larger overlap of the dominance functions in fast speech and hence of the speech gestures themselves. By varying the dominance functions, different ways of coarticulation can be simulated.

Figure 5 shows the articulation of some phonemes as produced by our muscle model and associated contraction values. In Figure 6, the position of the tongue during the pronunciation of a / ø / is depicted.

5.3 *Asynchronous Simulation and Rendering*

Our facial animation system efficiently exploits dual processor systems by using individual *threads*³ for the physics-based simulation and the rendering of an animation, see also Figure 1. In our implementation we use the `pthread` library as an interface to the POSIX thread model.

The *simulation thread* is controlled by muscle parameters from an animation script or from user interaction. An animation script can be automatically generated in a speech analysis preprocessing step (cf. Section 5.2). During the simulation, the Lagrangian equations of motion for a mass-spring system are numerically integrated through time using an explicit forward integration scheme. The structure of the mass-spring system is described in detail in [17]. After a simulation step has been computed, the resulting displacements of the face mesh vertices are stored in a *key frame* entry in a ring buffer. In addition, the current animation parameters of our facial components (see Section 4.1) are stored in the same key frame. The time steps of the simulation need not be constant, but can be adapted to the needs of speech synchronization. To ensure a temporally undistorted rendering of irregularly distributed key frames,

³subroutines running asynchronously on different CPU's of a multi-processor machine

the time stamp t_i of each simulation step (measured in wall-clock time) is also stored within the associated key frame f_i .

The *rendering thread* reads and interpolates key frames from the ring buffer and renders the face model according to the interpolated animation and displacement parameters. Depending on the “temporal distance” $t_{i+1} - t_i$ between two successive key frames f_i and f_{i+1} , we typically generate three to ten interpolated frames from the two key frames. The blending factor α_R , which is used to generate the interpolated frame $f_R = \alpha_R f_i + (1 - \alpha_R) f_{i+1}$, is computed from the time stamps of the key frames involved and the rendering time t_R ($t_i \leq t_R \leq t_{i+1}$): $\alpha_R = (t_{i+1} - t_R) / (t_{i+1} - t_i)$. If t_R becomes larger than t_{i+1} , the key frame f_i is discarded and the new key frame f_{i+2} is read from the ring buffer. Now the interpolation takes place between f_{i+1} and f_{i+2} , accordingly. The rendering time t_R is also measured in wall-clock time, but shifted by a certain delay to the simulation time. The amount of the delay depends on the time the simulation thread needs to compute the first two key frames of the animation.

To enhance performance, we do not use any locking mechanism for the ring buffer access. However, we must make sure that neither the rendering thread \mathcal{R} tries to read a key frame that has not yet been written by the simulation thread \mathcal{S} , nor that \mathcal{S} overwrites a key frame that has not yet been read by \mathcal{R} . This is accomplished by using a first-in-first-out (FIFO) buffer b with mutual access and one additional variable v that is locked by a mutex. Whenever \mathcal{S} has finished writing a new key frame, the index of that key frame (modulo the size of the ring buffer) is stored in the FIFO buffer b . If \mathcal{R} needs to read a new key frame from the ring buffer, the respective index is read from b . This read call blocks if b is empty. On the other hand, \mathcal{R} stores the index of the older key frame that is currently used in the variable v . Before writing to the ring buffer, \mathcal{S} checks the content of v . If the index j of the new key frame is equal to v , writing is delayed until $v > j$.

Our ring buffer access mechanism performs very well on dual processor operating platforms for a ring buffer size of about 10–20 key frames. Neither the simulation thread nor the rendering thread had to wait for ring buffer access. Giving \mathcal{S} a competitive edge of about half the size of the ring buffer minimizes the probability of blocking. In addition, our dynamic rendering approach presented in [16] can be employed within the rendering thread to improve the visual appearance of coarse base meshes especially in the region around the mouth.

5.4 Randomized Animation Parameters

To achieve a more lifelike appearance of the virtual head, we included eye blinking and small random head movements into the animation. To this effect,

activity	T_I	T_A
acquire range scans	15–20	—
acquire photographs	5–10	—
process range scans	250–350	45
prepare mesh for animation	30	1
register textures	30–60	< 5
fit skull & fine-tune muscles	20–45	< 1

Tab. 1: Time spent for diverse preprocessing activities. T_I denotes the amount of time that was spent interactively, while T_A indicates the additional amount of time that went into fully automatic computations. All timings are given in minutes.

the animation thread generates different scalar-valued noise functions that can be applied to animation parameters. We have used a function producing peaks in a randomized time interval which is applied to eyelid closure. Sine wave functions of different period and with randomized amplitude are applied to head rotation around the x -, y -, and z -axis. These effects add greatly to the realism of the animation by making the head appear much less static.

6 Results and Conclusion

We have presented a system for facial modeling and animation, which aims at the generation of high-quality models and animation with as little effort as possible. Our system includes several tools that facilitate mesh processing, texture registration, and assembling of skin, skull, muscles, and facial components (see Figure 7). Some movies demonstrating the capabilities of our system can be downloaded from the URL <http://www.mpi-sb.mpg.de/resources/FAM/>.

Table 1 shows how much time we typically spent on diverse preprocessing activities on the way from data acquisition to the final animatable head model. The acquisition and processing of range data takes considerable time and effort, mostly due to missing data that can't be obtained by the scanner. In particular, the ears are a problem as well as the inner part of the lips. Although mesh decimation results in a usable approximation of the input data, there is not enough control over the topology of the resulting mesh. This can lead to artifacts in the animation due to asymmetries between the left and right side of the face mesh and misalignment of the triangle edges. We are thus investigating fitting a previously modeled, generic head directly to the range scan data. Subdivision surfaces can be used to achieve adaptive resolution with well-known mesh topology.

Another step in the construction of the head model that introduces some manual work is registration of textures, which we would like to be fully automated. Also, modeling and rendering of hair needs to be included. Though typically no data for the hair-covered parts of the head can be obtained from range scanning devices, we are confident that geometry and texture of the haircut can be extracted from the acquired photographs.

In the context of speech synchronization, we found that the approach of COHEN and MASSARO [7] was straightforward to be adapted to our muscle-based model. Once the parameters of the dominance function and the target contraction parameters of each muscle have been interactively assigned to every phoneme, key frames for an animation synchronized to a labeled speech signal can be generated automatically. No video taping of the speakers and no neural network training is required.

The rendering performance of our system achieves real-time frame rates of about 100 fps (frames per second) on a dual processor Pentium 4 Linux-PC (2x 1.7 GHz) with a GeForce3 graphics board. On a 1 GHz single processor PC, we obtain frame rates of about 30–35 fps.

Acknowledgements

The author would like to thank all of the people involved in the development of the MEDUSA facial modeling and animation system: Kolja Kähler, Hitoshi Yamauchi, Irene Albrecht, Won-Ki Jeong, Jacques Koreman, and Hans-Peter Seidel. Without their effort, a complex system like MEDUSA could neither be developed nor maintained within a reasonable amount of time.

References

- [1] Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. Speech Synchronization for Physics-based Facial Animation. In *Proc. WSCG 2002*, pages 9–16, 2002.
- [2] A.-P. Benguerel and H. A. Cowan. Coarticulation of Upper Lip Protrusion in French. *Phonetica*, 30:41–55, 1974.
- [3] R. A. W. Bladon and A. Al-Bamerni. Coarticulation Resistance in English /l/. *Journal of Phonetics*, 4:137–150, 1976.
- [4] Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Computer Graphics (SIGGRAPH '99 Conf. Proc.)*, pages 187–194. ACM SIGGRAPH, August 1999.
- [5] John E. Chadwick, David R. Haumann, and Richard E. Parent. Layered Construction for Deformable Animated Characters. In *Computer Graphics (SIGGRAPH '89 Conf. Proc.)*, volume 23, pages 243–252. ACM SIGGRAPH, July 1989.
- [6] David T. Chen and David Zeltzer. Pump it up: Computer Animation of a Biomechanically Based Model of Muscle using the Finite Element Method. In *Computer Graphics (SIGGRAPH '92 Conf. Proc.)*, volume 26, pages 89–98. ACM SIGGRAPH, July 1992.
- [7] Michael M. Cohen and Dominic W. Massaro. Modeling Coarticulation in Synthetic Visual Speech. In *Models and Techniques in Computer Animation*, pages 139–156. Springer-Verlag, 1993.
- [8] Ron Cole, Dominic W. Massaro, Jacques de Villiers, Brian Rundle, Khaldoun Shobaki, John Wouters, Michal Cohen, Jonas Beskow, Patrick Stone, Pamela Connors, Alice Tarachow, and Daniel Solcher. New Tools for Interactive Speech and Language Training:

- Using Animated Conversational Agents in the Classrooms of Profoundly Deaf Children. In *Proc. ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education*, London, UK, April 1999.
- [9] Entropic Research Laboratory, Inc., Sheraton House, Castle Park, CBX 0AX Cambridge, UK. *ESPS Manual*, 1993.
- [10] J. S. Garofolo. *Getting Started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database*. National Institute of Standards and Technology (NIST), Gaithersburgh, MD, 1988.
- [11] Taro Goto, Marc Escher, Christian Zanardi, and Nadia Magnenat-Thalmann. MPEG-4 based Animation with Face Feature Tracking. In *Proc. Eurographics Workshop on Computer Animation and Simulation '99*, pages 89–98, 1999.
- [12] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Frédéric Pighin. Making Faces. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, pages 55–66. ACM SIGGRAPH, July 1998.
- [13] David R. Hill, Andrew Pearce, and Brian Wyvill. Animating Speech: An Automated Approach using Speech Synthesised by Rules. *The Visual Computer*, 3(5):277–289, March 1988.
- [14] Horace H. S. Ip and C. S. Chan. Script-Based Facial Gesture and Speech Animation Using a NURBS Based Face Model. *Computers & Graphics*, 20(6):881–891, November 1996.
- [15] ISO/IEC. Overview of the MPEG-4 Standard. <http://www.cse.lt.it/mpeg/standards/mpeg-4/mpeg-4.htm>, July 2000.
- [16] Kolja Kähler, Jörg Haber, and Hans-Peter Seidel. Dynamic Refinement of Deformable Triangle Meshes for Rendering. In *Proc. Computer Graphics International 2001 (CGI 2001)*, pages 285–290, July 2001.
- [17] Kolja Kähler, Jörg Haber, and Hans-Peter Seidel. Geometry-based Muscle Modeling for Facial Animation. In *Proc. Graphics Interface 2001*, pages 37–46, June 2001.
- [18] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. SMILE: A Multilayered Facial Animation System. In *Proc. IFIP WG 5.10, Tokyo, Japan*, pages 189–198, 1991.
- [19] Leif Kobbelt, Mario Botsch, Kolja Kähler, Christian Rössl, Robert Schneider, and Jens Vorsatz. Geometric Modeling Based on Polygonal Meshes. In *Eurographics 2000 Tutorial Notes*. Blackwell Publishers, 2000.
- [20] Rolf M. Koch, Markus H. Groß, and Albert A. Bosshard. Emotion Editing using Finite Elements. In *Computer Graphics Forum (Proc. Eurographics '98)*, volume 17, pages C295–C302, September 1998.
- [21] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Constructing Physics-based Facial Models of Individuals. In *Proc. Graphics Interface '93*, pages 1–8, May 1993.
- [22] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic Modeling for Facial Animations. In *Computer Graphics (SIGGRAPH '95 Conf. Proc.)*, pages 55–62. ACM SIGGRAPH, August 1995.
- [23] John P. Lewis and Frederic I. Parke. Automated Lip-Synch and Speech Synthesis for Character Animation. In John M. Carroll and Peter P. Tanner, editors, *Proceedings of Human Factors in Computing Systems and Graphics Interface '87*, pages 143–147, April 1987.
- [24] Anders Löfqvist. Speech as Audible Gestures. In W. J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modelling*, pages 289–322. Kluwer Academic Publishers, 1990.
- [25] Monique Nahas, Herve Huitric, and Michel Saintourens. Animation of a B-Spline Figure. *The Visual Computer*, 3(5):272–276, March 1988.
- [26] Frederic I. Parke. *A Parametric Model for Human Faces*. PhD thesis, University of Utah, Salt Lake City, UT, December 1974.
- [27] Frederic I. Parke. Parameterized Models for Facial Animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, November 1982.

- [28] Andrew Pearce, Brian Wyvill, Geoff Wyvill, and David Hill. Speech and Expression: A Computer Solution to Face Animation. In *Proceedings of the Graphics Interface '86*, pages 136–140, May 1986.
- [29] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing Realistic Facial Expressions from Photographs. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, pages 75–84. ACM SIGGRAPH, July 1998.
- [30] Stephen M. Platt and Norman I. Badler. Animating Facial Expressions. In *Computer Graphics (SIGGRAPH '81 Conf. Proc.)*, volume 15, pages 245–252. ACM SIGGRAPH, August 1981.
- [31] Claudio Rocchini, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Multiple Textures Stitching and Blending on 3D Objects. In *Rendering Techniques '99 (Proc. 10th Eurographics Workshop on Rendering)*, pages 119–130, 1999.
- [32] Ferdi Scheepers, Richard E. Parent, Wayne E. Carlson, and Stephen F. May. Anatomy-Based Modeling of the Human Musculature. In *Computer Graphics (SIGGRAPH '97 Conf. Proc.)*, pages 163–172. ACM SIGGRAPH, August 1997.
- [33] Demetri Terzopoulos and Keith Waters. Physically-based Facial Modelling, Analysis, and Animation. *Journal of Visualization and Computer Animation*, 1(2):73–80, December 1990.
- [34] Roger Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, June 1986.
- [35] Keith Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics (SIGGRAPH '87 Conf. Proc.)*, volume 21, pages 17–24. ACM SIGGRAPH, July 1987.
- [36] Keith Waters and Joe Frisbie. A Coordinated Muscle Model for Speech Animation. In *Proc. Graphics Interface '95*, pages 163–170, May 1995.
- [37] Jane Wilhelms and Allen Van Gelder. Anatomically Based Modeling. In *Computer Graphics (SIGGRAPH '97 Conf. Proc.)*, pages 173–180. ACM SIGGRAPH, August 1997.
- [38] Lance Williams. Performance-Driven Facial Animation. In *Computer Graphics (SIGGRAPH '90 Conf. Proc.)*, volume 24, pages 235–242. ACM SIGGRAPH, August 1990.
- [39] Yin Wu, Prem Kalra, Laurent Moccozet, and Nadia Magnenat-Thalmann. Simulating Wrinkles and Skin Aging. *The Visual Computer*, 15(4):183–198, 1999.
- [40] Yin Wu, Nadia Magnenat-Thalmann, and Daniel Thalmann. A Plastic-Visco-Elastic Model for Wrinkles in Facial Animation and Skin Aging. In *Proc. Pacific Graphics '94*, pages 201–214, August 1994.

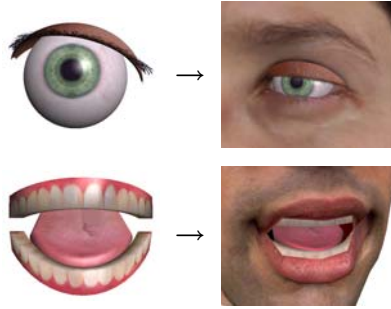
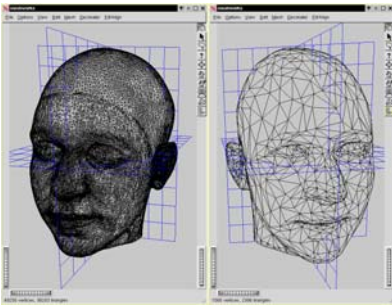


Fig. 8: Generic models of eyes, teeth, and tongue (left) are fitted into individual face meshes (right).

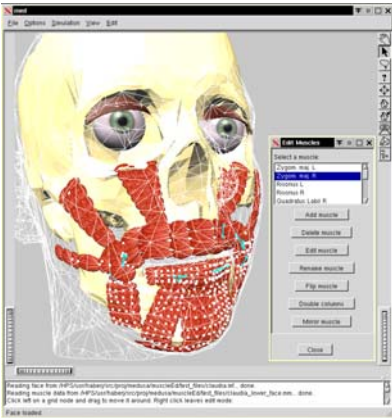


Fig. 7: Tools of our facial modeling and animation system. Top to bottom: a) Two views of our mesh decimation and alignment tool. b) Graphical user interface (GUI) of our texture registration tool. c) GUI of our muscle editor.

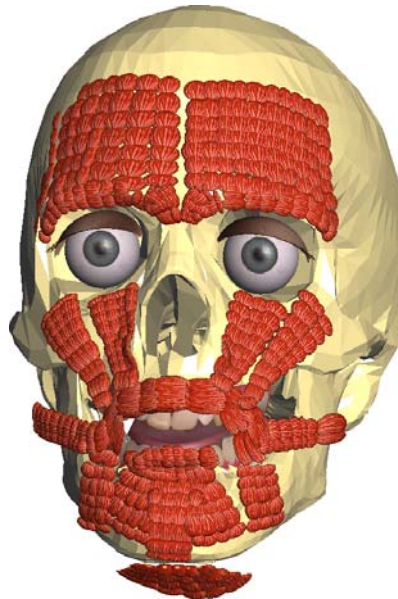


Fig. 9: Reference skull with parts: eyes, teeth, tongue, muscles.