A Head Model with Anatomical Structure for Facial Modeling and Animation

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

vorgelegt von

Kolja Kähler

Eingereicht in Saarbrücken am 19.5.2003

Tag der mündlichen Prüfung: 17.12.2003

1. Gutachter: Prof. Dr. Hans-Peter Seidel
2. Gutachter: Prof. Dr. Philipp Slusallek
Dekan: Prof. Dr. Philipp Slusallek

Abstract

In this dissertation, I describe a virtual head model with anatomical structure. The model is animated in a physics-based manner by use of muscle contractions that in turn cause skin deformations; the simulation is efficient enough to achieve real-time frame rates on current PC hardware. Construction of head models is eased in my approach by deriving new models from a prototype, employing a deformation method that reshapes the complete virtual head structure. Without additional modeling tasks, this results in an immediately animatable model. The general deformation method allows for several applications such as adaptation to individual scan data for creation of animated head models of real persons. The basis for the deformation method is a set of facial feature points, which leads to other interesting uses when this set is chosen according to an anthropometric standard set of facial landmarks: I present algorithms for simulation of human head growth and reconstruction of a face from a skull.

The creation of computer-animated human faces is a long-standing and challenging problem since the early 1970s. There are numerous approaches to facial animation, but to this day no general-purpose system exists that solves the problem in a manner satisfying the needs of all practical applications. In the medical field, highly accurate reproduction of a real head is required to enable well-informed decisions in surgery planning. Animation capabilities and computation time are not important. On the other hand, realism is of minor concern in interactive dialog systems or computer games. Here, the animation merely has to look plausible but play in real time. A real-life model does not need to be reproduced with all subtleties of facial shape and texture.

An obvious approach to achieve generality is the simulation of the inner workings of a real face. High expectations are tied to physics-based systems, where the ultimate goal is to have the full range of conformation and expressiveness in the face emerge "naturally" through precise modeling of the anatomical structure and accurate simulation of tissue properties. This has so far only been realized in parts, and no current implementation catches all the intricacies of the human face.

Traditionally, the computational cost of physics-based simulation has been prohibitive for real-time facial animation on consumer-class PC hardware. This has changed dramatically in recent years, making fast high quality animation possible on current desktop computers. But, apart from the run-time issues, constructing a virtual head model with the complex structure of skull, muscles, and skin is non-trivial, requiring artistic skills and time.

This situation motivates my dissertation: I propose an anatomy-based virtual head model that is animatable in real time using numerical simulation techniques, driven by an advanced facial muscle model. The simulation is efficient enough to achieve real-time frame rates on current PC hardware. Manual construction of such a structured head model is a difficult task, which is avoided in my approach by deriving new models from a prototype, employing a deformation method that reshapes the complete head structure. Without additional modeling tasks, this results in an immediately animatable model. The general deformation method allows for several applications such as adaptation of a reference head model to individual scan data to produce animated head models of real persons. The methods and techniques described are demonstrated on human head models, but they are also applicable to other virtual creatures, with few modifications in the construction process.

The deformation method is based on facial feature points, which leads to other interesting uses when an anthropometric standard set of facial landmarks is chosen: using a database of facial measurements, shape attributes of the face can be changed in a controlled, meaningful manner. As an example, I show how statistically plausible growth of an individual head from childhood to adult age can be simulated. Furthermore, based on known skull / skin distance relationships, a face can be approximated from the skull geometry. These applications demonstrate the usefulness of the approach outside the realm of computer graphics, for instance, in the forensic sciences.

In summary, the key contributions made to the field in this dissertation are:

- an **anatomy-based head model**: the human head's major structural components are built explicitly into its virtual counterpart: a flexible *skin surface*, an advanced model for *virtual muscles* controlling the animation, an embedded *skull* to which skin and muscles attach, as well as separately modeled geometric objects for *eyes, teeth*, and *tongue*. A *mass-spring system* connects skin, muscles, and skull. Additionally, a set of *landmarks*, defined on the skin and skull surfaces, is part of the model. The model and associated algorithms enable real-time, physics-based animation.

- a general **deformation method for the head model**: all parts of the virtual head structure are reshaped simultaneously using a landmark-based approach; the resulting model is immediately animatable with no further modeling steps.

- **creation of animatable head models from range scan data**: an algorithm is presented that employs the deformation method for adaptation of a generic head template to scan data.

- **simulation of human head growth**: making use of landmark-based anthropometric measurements, the shape changes of the human head from infancy to adulthood are simulated.

- **reconstruction of faces from skull data**: I propose a technique that allows fast and flexible modeling of face reconstructions from a scanned skull. This has applications mainly in the forensic sciences.

Zusammenfassung

In dieser Dissertation beschreibe ich ein ein nach der menschlichen Anatomie strukturiertes virtuelles Kopfmodell. Dieses Modell wird physikbasiert durch Muskelkontraktionen bewegt, die wiederum Hautdeformationen hervorrufen; die Simulation ist effizient genug, um Echtzeitanimation auf aktueller PC-Hardware zu ermöglichen. Die Konstruktion eines Kopfmodells wird in meinem Ansatz durch Ableitung von einem Prototypen erleichtert, wozu eine Deformationstechnik verwendet wird, die die gesamte Struktur des virtuellen Kopfes transformiert. Ein vollständig animierbares Modell entsteht so ohne weitere Modellierungsschritte. Die allgemeine Deformationsmethode gestattet eine Vielzahl von Anwendungen, wie beispielsweise die Anpassung an individuelle Scandaten für die Erzeugung von animierten Kopfmodellen realer Personen. Die Deformationstechnik basiert auf einer Menge von Markierungspunkten im Gesicht, was zu weiteren interessanten Einsatzgebieten führt, wenn diese mit Standard-Meßpunkten aus der Anthropometrie identifiziert werden: Ich stelle Algorithmen zur Simulation des menschlichen Kopfwachstums sowie der Rekonstruktion eines Gesichtes aus Schädeldaten vor.

Die Erzeugung computeranimierter menschlicher Gesichter ist ein bereits seit den 70er Jahren bestehendes und nach wie vor herausforderndes Problem. Es gibt eine Vielzahl von Ansätzen für Gesichtsanimation, aber bis heute existiert kein universell nutzbares System, das diese Problem auf eine Art löst, die für sämtliche praktischen Anwendungen brauchbar ist. In der Medizin ist die präzise Reproduktion des realen Kopfes unumgänglich, um gut fundierte Entscheidungen in der Operationsplanung treffen zu können. Animation und Rechenzeit sind hier nicht von großer Relevanz. Andererseits ist Realismus nur von minderer Bedeutung, wenn es um interaktive Dialogsysteme oder Computerspiele geht. In diesem Fall muß selbst eine reale Vorlage im Modell nicht mit allen Feinheiten der Gesichtsform und -textur reproduziert werden.

Eine naheliegende Herangehensweise für ein universell einsetzbares System ist die Simulation der internen Zusammenhänge eines realen Gesichtes. Große Erwartungen werden hier an physikbasierte Systeme geknüpft, wobei das hochgesteckte Ziel das “natürliche” Hervorbringen des kompletten Spektrums menschlicher Gesichtsregungen und -formen ist, erzielt durch präzise Modellierung der anatomischen Struktur und genaueste Simulation der Gewebeeigenschaften. Dies ist bislang nur in Teilen realisiert worden, und keine aktuelle Implementierung wird der Komplexität des menschlichen Gesichts wirklich gerecht.

Traditionell ermöglichten die zu geringen Rechenkapazitäten keine Echtzeit-Gesichtsanimation mittels physikbasierter Simulation auf handelsüblicher PC-Hardware. Dieser Zustand hat sich in den letzten Jahren dramatisch verändert, wodurch hochqualitative Animation auch auf Desktopcomputern möglich wird. Jedoch ist, unabhängig von diesen Laufzeitproblemen, die Konstruktion eines virtuellen Kopfmodells mit der komplexen Struktur von Schädel, Muskeln, und Haut ein schwieriges und zeitaufwendiges Unterfangen, das künstlerische Fähigkeiten voraussetzt.

Diese Situation motiviert meine Dissertation: Ich schlage ein anatomiebasiertes virtuelles Kopfmodell vor, das unter Verwendung numerischer Simulationsmethoden in Echtzeit animierbar ist, angetrieben durch ein neuartiges Muskelmodell. Die manuelle Konstruktion eines solchen strukturierten virtuellen Kopfes ist eine schwierige Aufgabe, die in meinem System durch Ableitung neuer Modelle von einem Prototypen vermieden wird, unter Verwendung einer Deformationstechnik, die die gesamte Kopfstruktur verformt. Ohne zusätzliche Arbeitsschritte resultiert daraus ein sofort animierbares Modell. Das generelle Deformationsprinzip gestattet vielfältige Anwendungen, wie beispielsweise die Anpassung eines Referenzkopfes an individuelle Scandaten, um animierte Kopfmodelle realer Personen zu generieren. Die in dieser Arbeit beschriebenen Techniken werden an menschlichen Kopfmodellen demonstriert, sind jedoch mit wenigen Änderungen im Konstruktionsprozeß ebenso auf künstliche Kreaturen anwendbar.

Die Deformationstechnik basiert auf im Gesicht angebrachten Markierungspunkten, was zu weiteren interessanten Anwendungen führt, wenn diese mit Standard-Meßpunkten aus der Anthropometrie identifiziert werden: Mittels einer Datenbank von Gesichtsmessungen können formgebende Attribute des Gesichtes in sinnvoller, kontrollierter Weise verändert werden. Als konkretes Beispiel stelle ich die Simulation des individuellen Kopfwachstums von der Kindheit bis zum Erwachsenenalter vor. Weiterhin kann ein Gesicht aus der Geometrie des Schädels unter Kenntnis der Abstände zwischen Haut und Knochen näherungsweise abgeleitet werden. Diese Anwendungen demonstrieren die Nützlichkeit des vorgestellten Ansatzes auch außerhalb der Computergraphik, wie beispielsweise in der Forensik.

Im folgenden sind die wesentlichen Beiträge dieser Dissertation noch einmal zusammengefaßt:

- Ein **anatomiebasiertes Kopfmodell**: Die wichtigsten strukturellen Komponenten des menschlichen Kopfes sind explizit in das virtuelle Gegenstück integriert: Eine flexible *Hautoberfläche*, ein neuartiges Modell für *virtuelle Muskeln*, die die Animation steuern, ein eingebetteter *Schädel*, an den Haut und Muskeln anknüpfen, sowie separat modellierte geometrische Objekte für *Augen, Zähne* und *Zunge*. Ein Masse-Feder-Netzwerk verbindet Haut, Muskeln und Knochen. Zusätzlich ist eine Menge von *Markierungspunkten* auf der Oberfläche von Haut und Schädel Bestandteil des Modells. Das Modell und die korrespondierenden Algorithmen ermöglichen physikbasierte Echtzeitanimation.

- Eine universelle **Deformationstechnik für das Kopfmodell**: Die gesamte virtuelle Kopfstruktur wird mittels einer auf den Markierungspunkten beruhenden Technik simultan verformt; das resultierende Modell ist ohne weitere Arbeitsschritte sofort animierbar.

- **Erzeugung animierbarer Kopfmodelle aus Scandaten**: Ein Algorithmus wird vorgestellt, der die Deformationstechnik zur Anpassung eines generischen Kopfmodells an Scandaten verwendet.

- **Simulation menschlichen Kopfwachstums**: Aufbauend auf anthropometrischen Messungen, die ebenfalls Markierungspunkte im Gesicht verwenden, werden die Formveränderungen des menschlichen Kopfes vom Säuglingsalter bis zum Erwachsenen simuliert.

- **Rekonstruktion von Gesichtern aus Schädeldaten**: Die von mir vorgestelle Technik ermöglicht schnelle und flexible Erzeugung von Gesichtsrekonstruktionen auf von einem 3D-Scanner erfaßten Schädeln. Anwendungen hierfür finden sich vor allem in der Forensik.

Acknowledgements

For their support and assistance in the work presented in this thesis I am greatly indebted to many people. First and foremost, I owe a special debt to my thesis advisor, Professor Hans-Peter Seidel, for his encouragement and support, and for making it possible for me to carry out my research at the MPI Informatik in Saarbrücken. His efforts in building up the computer graphics group have resulted in a very productive and stimulating work environment, which I greatly enjoyed being a part of. I am also grateful to my external reader, Professor Philipp Slusallek, who expressed avid interest in this work, and found the right words at the right time.

I thank all my colleagues at the MPI for making my stay worthwhile and fun from both a scientific and personal point of view. I am especially indebted to the members of the facial modeling and animation group, in particular to Dr. Jörg Haber, who never stopped pushing for that "extra bit" in all areas of research. He was an unlimited resource of encouragement, constructive criticism, and support, as well as a companion during the many late hours that went into this work. Irene Albrecht valiantly endured the shortcomings of my early experimental animation system when using it for her own work on speech animation. Her invaluable feedback sparked the inspiration for many of the features of the system finally described here. I am also grateful to these two people for proofreading early versions of this text. Hitoshi Yamauchi, who never sleeps, contributed much to the look of the various faces shown on the following pages through his work on texture mapping. A fruitful combination with Won-Ki Jeong's research on subdivision surface construction resulted in the section on animated subdivision surface head models.

Several individuals outside the MPI deserve special mention. For their active support and valuable insights regarding generation of synthetic human speech, I wish to thank Dr. Jacques Koreman and Marc Schröder. I also owe thanks to Dr. Dieter Buhmann and Daniela Bellmann for showing enthusiasm about the facial reconstruction aspect of my work, as well as for helpful comments and provision of materials.

Finally, I wish to thank Mario Botsch, Leon Kästele, Claudia Skera, and Jens Vorsatz for allowing me to acquire and use range scans of their heads.

To Claudia, for be(ar)ing with me.

Contents

List of Figures

CHAPTER 1

Introduction

> *I never forget a face, but in your case I'll be glad to make an exception.*
> *– Groucho Marx*

Every single human face is unique – in its look, its shape and proportions, as well as in the specific ways of articulation and expression that are so characteristic for an individual. This simple fact is both the blessing and the curse of computer-generated facial animation. It's a blessing because the human perceptual system can deal with a wide range of different faces in everyday life. The acceptable shape variations thus allow for a large number of possible virtual faces, making it easy to create a face that has a recognizable, unmistakable appearance. In the hands of a skilled animator, vast distortions of shape and exaggerated timing of motion can even *increase* the perceived realism of an animation, as LASSETER has shown in his seminal paper on the principles of animation [Las87]. However, the uniqueness of faces can also be a curse: the efficiency of the perceptual system enables people to disambiguate faces with extremely subtle differences. This makes it hard to create a virtual face that convincingly resembles a real person, especially someone familiar. A deviation in even the tiniest nuance in the shape, texture, or motion of the face is immediately recognized as wrong.

The creation of computer-animated human faces is a long-standing and challenging problem since the early 1970s. As PARKE has noted [Par82], the more realistic animated faces are, the more critical and less forgiving the human observer becomes. Early face models could be clearly recognized as being artificial, while the degree of realism that can be achieved today often leaves the observer with an uneasy feeling of being "tricked" into a flawed illusion. From this point of view, facial animation has actually become even *harder* with the advances that have been made over the years.

There are numerous approaches to facial animation, all struggling with the enormous complexity of the human face in geometry and in motion. To this day, no general-purpose system exists that solves the problem in a manner satisfying the needs of all practical applications. In the medical field, highly accurate reproduction of a real head is required to enable well-informed decisions in surgery planning. Animation capabilities and computation time are not important. On the other hand, realism is of minor concern in interactive dialog systems or computer games. Here, the animation merely has to look plausible, but play in real time. Even a real-life model does not need to be reproduced with all subtleties of facial shape and texture.

An obvious approach to achieve generality is the simulation of the inner workings of a real face. High expectations are tied to physics-based systems, where the ultimate goal is to have the full range of conformation and expressiveness in the face emerge "naturally" through precise modeling of the anatomical structure and accurate simulation of tissue properties. PLATT and BADLER put this into words in 1981 [PB81]:

> "Analysis and simulation of the face are based on the actual structure of the face. Therefore, any constraints or peculiarities of the real face should appear within the system."

This vision has so far only been realized in parts, and no current implementation catches all the intricacies of the human face. But the potential for continuous improvement is huge, once an anatomy-based framework is established: with medical progress, more knowledge about human anatomy will be acquired, leading to more accurate models of the structure of the human head and of facial dynamics. Advances in numerical algorithms and more powerful hardware will lead to more accurate simulations and thus more convincing animations.

1.1 Motivation

Traditionally, the computational cost of physics-based simulation has been prohibitive for generation of real-time facial animation on consumer-class PC hardware. This has changed dramatically in recent years, making fast high quality animation possible on current desktop computers. But, apart from the run-time issues, constructing a virtual head model with the complex structure of skull, muscles, and skin is non-trivial, requiring artistic skills and time. This situation provides the motivation for my thesis: I propose an anatomy-based virtual head model that is animatable in real time using numerical simulation techniques, driven by an advanced facial muscle model. To reduce the cumbersome work of model construction, a deformation technique is presented that allows, for instance, adaptation of the complete head model structure to scan data as well as other modifications of head shape. All created models are instantly animatable, unifying the usual steps of modeling and "rigging for animation". The animation parameters are shared between models, thus enabling re-use of animation scripts.

The structural deformation technique takes advantage of existing anthropometric knowledge. Using a database of facial measurements, shape attributes of the face can be changed in a controlled, meaningful manner. As an example, I show how statistically plausible growth of an individual head from childhood to adult age can be simulated. Furthermore, based on known skull / skin distance relationships, a face can be approximated from the skull geometry. These applications demonstrate the usefulness of the approach outside the realm of computer graphics, for instance, in the forensic sciences.

1.2 Notation

Following is a list of the more important notational conventions in the text, most of which relate to the mathematical parts.

- Scalars are written in italics, using upper and lower case Latin and Greek letters: a, b, C, λ, μ. Integral quantities such as "the number of elements N" are always denoted in upper case letters.
- Names of scalar functions are written in Roman face: $\mathrm{f}(\theta)$
- Vectors, points, and vector-valued functions are written in lower case, bold letters: $\mathbf{a} \in \mathbb{R}^3, \mathbf{x}(t)$.
- Vectors are column vectors by default: $\mathbf{x} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = (a\ b\ c)^T$.
- To reduce clutter, instead of writing e.g. $\mathbf{x}(t)$, the short form $\mathbf{x}$ is sometimes used, when the function argument is clear from the context, or $\mathbf{x}^t$ when t can assume only discrete values.
- Matrices are written in upper case bold letters: $\mathbf{A}, \mathbf{B}, \mathbf{C}$.
- Sets are written in upper case calligraphic letters: $\mathcal{A}, \mathcal{B}, \mathcal{C}$.
- $\dot{\mathrm{x}}(t)$ and $\ddot{\mathrm{x}}(t)$ denote the first and second derivative of x, respectively.
- Physical units are written in Roman font: kg, m, s.
- Code fragments are set in `typewriter font`.

CHAPTER 2

Applications and Expectations

> *I can think of nothing more boring for the American people than to have to sit in their living rooms for a whole half hour looking at my face on their television screens.*
>
> *– Dwight D. Eisenhower*

Before delving into the technicalities of facial animation, a thorough look at the range of applications for virtual faces is in order. As the quote above indicates, there should be more to it than just putting a face on the screen. In real life, human faces play a major role in visual communication, especially in social contexts: we look at, listen to, and talk with many real faces everyday. In addition, human faces are depicted in drawings, in the movies, or sculpted in stone. Possible applications for facial modeling and animation can be seen in all these real-life scenarios, although, many ideas might seem far-fetched considering the current state of technology. Still, facial modeling and animation on the computer is already employed for applications that could not be realized without its computational power and visualization capabilities. Given the virtually endless potential uses for virtual faces, the attempt to assemble an exhaustive list is probably futile. Instead, I will focus on what seems to be the common ground for all application domains: visual communication. To clarify the different practical requirements, I have picked a few examples—partly already realized, partly still in research—and grouped them according to these modes of visual communication:

Examination: Here, the virtual face is merely being looked at in terms of recognition or aesthetic judgment. While facial dynamics can play a major role, for instance, in recognizing a face, most existing applications here deal with static face models.

Monologue: A virtual head is talking to the user, relating information, and possibly also emotions, via the face.

Man / machine dialog: A virtual character and the user interact. This requires a back-channel from the user to the computer to enable a proper reaction.

Immersive dialog: In an environment, where representations of real users and virtual characters coexist, different dialog configurations arise and often change in a dynamic manner.

This classification should allow for some general insights into important elements of successful uses of virtual faces. In the more complex dialog scenarios, the consideration of human communications skills becomes crucial: a dialog is always expected to follow certain rules. If there is a mismatch between the signals and expected reactions of one or both of the peers, communication can be severely disturbed. This chapter is also about such expectations.

2.1 Examination: Modeling and Manipulating Faces

To be able to observe a virtual face, it obviously has to be modeled first. Acquisition of real-world data and modeling operations on the head geometry are the main concerns in this category. Apart from purely artistic modeling, there are also important applications in such vital fields as criminology and medicine, which I will discuss in the following sections.

2.1.1 Criminology

A large part of law enforcement work deals with construction, identification, and modification of facial images and models. To this day, this work is largely carried out with the help of skilled artists. For instance, a crime victim describes facial features of the felon to an artist, who draws and improves an image, until the result looks close enough to the victim's mental image. A variation of this approach uses photographs, where templates for the three major facial regions are combined. This is also easily done with 2D images on a computer. Given a 3D face model with a suitable parameterization to modify the layout of facial features, this approach can be further refined and improved, as has been suggested by PARKE [Par82]. The issue of facial parameterization will be picked up again in the next chapter.

When a person, especially a child, went missing, there can be large changes in appearance over time, which may prevent identification due to outdated photographs. It is thus important to modify attributes of the face in the picture, such as the haircut as a simple case. More difficult is the simulation of age changes, which is usually done by an artist with the help of image processing tools on the computer. A lot of knowledge about anthropometric facial measurements and development during growth is needed here. This kind of processing can also be performed on a 3D model, as will be discussed in Chapter 8.

Lastly, the remains of a deceased person may consist of nothing more than a few bones and a skull. For identification, it is common practice to manually model the tissue layers on top of the skull in clay, using assumptions about tissue thickness, age, weight, and other physical attributes that can be derived from the material at hand. Reconstructing the face in this way, it is often possible to identify the person, and a comparison with pictures sometimes shows a surprisingly good match. A virtual counterpart of this face reconstruction technique will also be discussed in Chapter 8.

Forensic illustration as discussed above is still considered an art form as much as a scientific endeavor. A detailed study of the field can be found in TAYLOR's textbook [Tay01].

2.1.2 Medicine

In the medical field, use of and research on virtual head models concentrates mainly on pre-operative planning, for instance, in plastic surgery. Successful predictions of the effects of an operation critically depend on an accurate model of the head, including the material properties of hard and soft tissues. A realistic model requires reliable acquisition of detailed, volumetric patient data. Volume scanners have limited spatial resolution, depending on their type (common techniques are CT, MRI, and PET), and the different scanning techniques also vary in the type of tissue they are suited for. It may thus be necessary to combine several scans into one data set. Given enough data, for a working model of the head the functional components must be extracted, such as skull, muscles and skin. To determine the outcome of a surgical operation, highly accurate simulation of the behavior of facial skin, the bone structure, and other tissues must be achieved.

From data acquisition over building the model to the simulation, many approximations are inevitably made, leading to a significant amount of uncertainty in the final system. For clinical applications, validation of simulated results is thus of prime importance—only if the results can be trusted to be similar to the actual surgery outcome, the simulation can be helpful in practical planning. These great demands on accuracy pose a number of hard research problems and involve a large amount of computations. In contrast, run-time speed of the simulation is not critical, although interactivity enables usage in the operating room.

2.2 Monologue: Talking Heads

In the previous examples the virtual head model does only communicate its shape and overall appearance, and remains mostly static. If content is to be brought to the observer, the need for animation arises: a face can talk to us by conveying emotions and through the spoken word.

2.2.1 Movies

In recent years, an increasing number of completely computer-generated full-feature movies have appeared in theaters. As much as in real-world movies, the faces of actors play a crucial role in the creation of a rich atmosphere that carries the audience along. The somewhat magical quality of *suspension of disbelief* is still the cornerstone of a successful animation. Not only high quality rendering, but also meticulous staging and timing are essential here. For instance, the much-discussed 2001 feature “Final Fantasy: The Spirits Within” [SM01] shows perfect rendering and modeling, including the actors’ faces. Still, the characters appear lifeless in animation, compared to a technologically outdated film like “Toy Story” [Las95], despite the decidedly non-realistic look of the characters in the latter. The requirements on facial animation are in either case very similar: artistic freedom in sculpting and animating the face is imperative, even more so in cartoon-like animation or fantasy films. Facial expressions are often manipulated on a frame by frame basis, to get exactly the right timing and motion needed by the artist to convey the message of a scene. It is typical to work in a hierarchical manner, i.e., to rough out the basic timing and flow of a motion sequence first, and then add the details. High quality rendering is very important, where much work goes, for example, into realistic shading of skin and hair. This becomes increasingly critical the more the characters are supposed to look like real human beings: faces are often shown in close-ups, where subtle inconsistencies in appearance or facial expression can destroy the suspension of disbelief. This type of animation is usually rendered over weeks on render farms consisting of possibly hundreds of CPUs.

2.2.2 The Virtual News Person

This is a classic application for facial animation. News-reading to an audience over the television screen has been conceived very early as an example use for talking heads: usually, news are read in a neutral fashion, so emotional, expressive animation is not predominant. More important is convincing speech synchronization [HPW88, LP87, BS83], which also becomes easier with language that is not tinged by strong emotions. Furthermore, the face is typically held in a relatively static position, facing the viewer frontally. Avoiding head and camera movements makes image-based 2D animation possible in this scenario. While the animations can be pre-rendered and played back later on, the ubiquity of the Web has lead to a new breed of *streaming animations*: animation parameters are delivered continuously to the user's browser, where a small application interprets them and renders the final image in a window. Compared to streaming rendered images, this method reduces the necessary bandwidth to just a fraction. Here, the question of a suitable parameterization reappears: it should be powerful enough to allow efficient encoding, but the user-side application has to run stably with low resources, so complex computations to translate the parameters into rendered animation frames are not feasible. One approach to streaming facial animation has now been standardized in the MPEG-4 specification [Koe02], enabling independent production of content and playback applications, which have usually been tied together by proprietary protocols and data formats.

2.3 Man / Machine Dialog: Responsive Characters

Successful communication between a human being and a machine is a challenging goal, and the size of the problem has led to the rise of a dedicated research field: human-computer interaction (HCI). Important in the context of facial animation is the dialog between a virtual face and a real user, or the dialog between users mediated by virtual faces. One of the main issues in any such dialog setting is the lack of information that the computer has about user actions, making it hard to generate proper responses.

2.3.1 Games

Computer games share some of the qualities of animated feature films, as they are telling a story and attempt to create an immersive setting. In real-time games, the highest priority lies on the interactivity and the smoothness of animation. In contrast to film, the quality of facial animation in a game is thus driven by the available hardware resources, an important factor being the graphics board, which is used for accelerated rendering. Given the technological progress in both hardware and software, game graphics are constantly improving, and it is conceivable that cinematic quality will be reached in the near future. A trend that can already be observed is the increasing replacement of pre-rendered cut scenes with scripted animations played back with in-game graphics. Games often use computationally cheap, relatively simple mechanisms on coarse polygon models to achieve real-time facial animation.

2.3.2 Dialog-Based Interfaces

As of today, interfacing to a computer is far from resembling a normal dialog situation. Using an automated teller machine, for instance, is very different from talking to a clerk at the bank. In dialog-based applications, it is important to consider human expectations pertaining to the mechanisms of a conversation, such as proper signaling and reaction to signals. Correct interpretation of the user's input by the computer is highly dependent on the machine's input communication channels. Currently, most computers employ the desktop metaphor to provide a virtual work environment. The limitations become obvious when a lot of information needs to be exchanged—typically by means of clicking dialog window buttons and entering text, since the input channels of a desktop computer are limited to a keyboard and a mouse. The search for other, more natural and richer means of communication, has led to the advent of speech reproduction and recognition technologies. Research on *conversational interfaces* promises to make such speech interfaces more lifelike and believable [Lai00]. Studies indicate that multi-modal interfaces, which support the audio with an animation of a talking head, can enhance the dialog. On the other hand, an interface incorporating audio and visual images can lead to far more acceptance problems than a purely text-based system: if the illusion of human behavior cannot be held up convincingly, feelings of disbelief, deception, and boredom can easily appear.

Considering not only faces, but complete characters, experiments with *embodied conversational agents* [CBB$^+$99, CB00] have shown that the interaction between man and machine can be made not only more pleasant, but also more effective by letting the computer mimic typical human communication behavior. Among the conversational skills an agent employs to create an atmosphere of trust are referrals to earlier encounters, exposing details about oneself, and being a good listener. With a complete character, it is also very important to keep body language and facial expressions consistent.

2.4 Immersive Dialog: Inhabited Virtual Worlds

Computer-controlled characters and representations of real users, known as *avatars*, coexist in virtual environments. Two or more virtual characters can be engaged in a dialog among themselves, a virtual character and a user may talk with each other, or a group of users may communicate through their virtual representations. STEPHENSON has developed a futuristic vision of the avatar scenario in his influential novel "Snow Crash" [Ste00], where he describes a nearly seamless integration between real and virtual worlds. So far, only very limited variations of the theme have found a place in reality, two of which I will discuss here.

2.4.1 Collaborative Work Environments

A group of distant co-workers can be brought together for working on a project using a shared virtual space. This is essentially an extension to teleconferencing, where the participants are represented by avatars in the virtual space, and they may not only communicate to exchange ideas, but also jointly access documents or objects in the environment. Most naturally, each person is represented with his or her own face, which raises the question of how an acceptable reproduction of an individual face can be achieved without too much effort. Ease of use is critical in these applications, since an overly complex setup or awkward controls may lead to rejection of the system. A very difficult problem is the credible reproduction of a person's typical expressions and gestures. In addition, the networked nature of a virtual environment raises practical issues: animations must be transferred to and displayed on a number of workstations, if possible in real time. Compression techniques and efficient coding of geometry and animation are vital here. This is more complex than in the news person scenario discussed before, since the environment is highly dynamic: natural communication between even only two participants through their virtual face representations has to be very quick and accurate to avoid latencies and misinterpretations that would give the dialog an awkward feeling.

2.4.2 Social Spaces

Collaboration in a group of people is an activity that is often more influenced by personal feelings and social issues than by the task at hand. Thus, collaborative environments have much in common with virtual social spaces, the graphical equivalent of text-based chat rooms. Focussed on suitability for home use, the demands and restrictions on such systems are even greater. Data acquisition for capturing individual appearance has to be cheap, simple, and fast. On home desktop computers, quality losses in rendering and animation are acceptable in favor of interactive execution, requiring adaptation to local computational resources. The quality threshold is at the level of recognizability, but in a pure social context the vanity aspect should not go unmentioned: people want to look good even in virtual reality. Of course, fantasy models and famous actors' faces also have their part in this, for anonymity or the fun of masquerade.

2.5 Summary

As can be seen from the above examples, the existing and potential uses for facial modeling and animation span a wide range of scientific and artistic topics. Depending on the application context, realization of a facial animation system requires knowledge from such diverse fields as story-telling, psychology, biology, physics, medicine, design, and of computer science. In the computer graphics domain, the challenges include problems from geometric modeling, rendering, animation, numerical simulation, interaction techniques, and so forth. The system requirements vary greatly with the targeted application. The variety of facial animation techniques currently in existence are examined in the next chapter.

CHAPTER 3

State of the Art

> *We are continually faced with a series of great opportunities*
> *brilliantly disguised as insoluble problems.*
> *– John W. Gardner*

Bringing animated faces to the computer screen continues to be an active area of research since PARKE published first results on the topic three decades ago [Par72]. While impressive progress has been made in the meantime, both technologically and methodically, the problem can still not be filed as solved. The approaches to modeling and animation of faces documented in the literature are diverse, as many sub-problems specific to the various application areas have to be dealt with. TERZOPOULOS and WATERS have described the process of facial animation for their system [TW90] as going through a hierarchy of abstractions, translating very high-level animation recipes into rendered images. Generalizing from this description, a common process for facial animation can be laid down in five stages:

1. **Expression:** Facial expressions and / or phonemes of specific intensity and duration are specified.
2. **Control:** Expressions are converted to a set of animation parameter values. Phonemes are mapped to their visual counterparts, visemes, which can be expressed as animation parameters.
3. **Simulation:** The facial animation parameters are interpreted, potentially performing complex computations based on an internal head model. The model may incorporate knowledge about, for instance, human anatomy or facial dynamics.
4. **Geometry:** The geometric model of the head is updated according to the results of the simulation. Parts of an animation can be carried out purely on the geometry level, by interpolating between neighboring simulation results.
5. **Images:** The geometry is finally rendered.

The complexity of each of these stages varies greatly with the particular animation approach. For instance, the simulation and geometry processing stages for an algorithm based on 2D images might reduce to selection of the right image to display. On the other hand, in a full-fledged physics-based system, the simulation part can be extremely complex. Since model properties and animation technique are tightly coupled, most published methods propose integrated solutions for modeling and animation, often combining several old and new approaches. In this survey of the existing body of work I will attempt to discuss the different problem areas one by one, which may lead to several mentions of some publications due to their integrated nature. To keep the size of this chapter manageable, I restrict here to rather general topics and issues of facial animation. Research that is relevant to specific sub-problems treated in this dissertation, for instance, on face aging, or numerical integration, is discussed in the corresponding chapter.

3.1 Animation of 2D Images

The standard graphics-enabled workstation uses a flat raster image for display. An obvious approach to animation is thus to stick to images without making the detour over construction and rendering of a geometric model. Animation from input photographs generally requires 2D morphing, usually based on the specification of features on the face image that is to be animated [BN92]. Subtle changes in detail that in reality accompany facial expressions are not included in such a warped image. LIU *et al.* [LSZ01] capture the changes in illumination on one person's face when making an expression, and map these details onto another, possibly warped, photograph. While this method shows substantial improvement on morphs of still images of faces, it is not directly usable for animation. A hybrid between image-based and geometry-based methods is the DECface system [WL93], which maps a frontal photograph onto a 2D polygon model comprised of 200 polygons.

Instead of still pictures, filmed sequences can also serve as input to a facial animation system. BREGLER *et al.* [BCS97] stitch together images from video footage of a speaking actor to create another video that shows the actor mouthing new utterances. New frames are composited from the existing material, choosing the best available match if a particular phoneme can't be exactly visualized. Building on the same idea, EZZAT *et al.* [EGP02] use a multidimensional morphable model to synthesize new mouth configurations not present in the input data, leading to improved results. BRAND *et al.* [Bra99] use input video footage with audio as training data to generate facial animation parameters from new audio sequences that can be used for warping 2D images as well as for driving geometric models.

A big advantage of image-based methods is their conceptual simplicity: if a photograph or video footage is used, the result is inherently photo-realistic, and no modeling of the different parts of a head is required. The price is the lack of freedom both for the viewer and the model itself: a change of posture that requires appearance of features invisible in the input, such as the teeth or wrinkles, cannot be done without further modeling. Likewise, a change in lighting conditions or realistic embedding into a three-dimensional environment is not possible. Applications for this kind of animation lie in user-interface agents, proper lip sync in dubbed movies, or the virtual news person.

3.2 3D Head Modeling

Due to the limitations of two-dimensional image manipulation techniques, most of the existing work on facial animation and modeling focuses on three-dimensional animated geometry. Modeling and animation are often closely related, because the animation techniques typically require a certain model structure. Depending on the approach, this structure can be highly complex. Modeling concentrates mainly on representation of the skin surface, since the skin comprises the largest visible area of the face, and also is subject to large deformations. Nonetheless, the other parts of a head such as eyes and teeth need to be modeled, too. A three-dimensional geometric model is in practice limited in resolution, and the reproduction of subtle detail, notably furrows and wrinkles, is often problematic. Commonly, geometric head models are thus improved by textures.

The most suitable geometric representation and modeling method for a head model depends on the application context and quality requirements: is a precise reproduction of an existing human face needed, a synthetic human face, or a fantasy creature? Does the animation run in real time or is it precomputed? Especially in interactive or real-time approaches, triangle meshes or quad meshes are common, while for high quality rendering spline surfaces and subdivision surfaces are often used.

3.2.1 Modeling Synthetic Human Faces

Head models that have no counterpart in reality are usually prepared manually, using commercial modeling packages. Another option is physical sculpting of a prototype, which is then digitized. For synthetic humans, the possible and plausible shape variations are limited, motivating methods that modify a generic head model. PARKE's [Par82] polygonal face model has several conformational parameters to adjust the size, shape, positioning, and color of the various facial features. As PARKE notes, there is little theoretical foundation for the chosen set of parameters, and he doesn't claim it to be ideal or even complete. The mesh for his face model is hand-tuned for proper animation, in so far as more polygons are inserted in areas of strong deformation, and edges are carefully aligned to facial features to control wrinkling of the surface. Another procedural approach to synthesis of plausible synthetic human heads with realistic proportions has been presented by DECARLO *et al.*[DMS98]. The head geometry is represented by a B-spline surface, which is deformed according to anthropometric constraints, taken from existing statistical data. The method is demonstrated on static models with no animation capabilities.

3.2.2 Modeling Real People

The bulk of the work in facial modeling focuses on replication of heads of real people. This requires acquisition of skin geometry either from photographs, 3D range scanners, or volume scanners. Generally, increased accuracy comes at the expense of substantially increased hardware cost, and requires longer and more complex acquisition procedures.

Apart from the skin, important parts of a complete, animatable head model are eyes, teeth, and tongue. These parts are typically modeled as generic objects which have to be aligned to the head model geometry [Par82, LTW95, MGR00]. Though advanced models for the tongue and the lips have been proposed [GMTA$^+$96, KPO00, KP00], their shape and appearance are usually not acquired from real persons.

Photographs

Much research has gone into generation of 3D models from photographs as a cheap and widely available data source. BLANZ *et al.* [BV99] use an extensive database of scanned heads, which was built up from 200 subjects by acquiring face geometry with a range scanner. From the data, a *face space* is derived by statistical analysis, where each point in the space represents a face. Given a photograph, an algorithm searches for the face shape and texture that best resembles the portrait. The generated static model is a dense, textured mesh of scan resolution. PIGHIN *et al.* [PHL$^+$98] deform a generic mesh model to adapt it to photographs taken simultaneously from multiple uncalibrated views, in practice using five cameras. In all the views, a set of feature points corresponding to a subset of the face mesh vertices has to be interactively marked. Given the correspondences, the pose of the head and the 3D positions of the marked features are recovered, and the deformation of the prototype mesh is computed by scattered data interpolation. The resulting model is textured using the same photographs. In a simpler variant of this approach, a coarse polygonal head model is constructed from a frontal and a side view of a person [LMT00, LWMT99], likewise deforming a generic polygon mesh prototype.

Range Scans

The geometry of a face can be acquired with much higher resolution and accuracy using range scanning devices. A detailed introduction into the procedure and problems of data acquisition using range scanners cannot be given here. An overview of common techniques can be found in KOBBELT *et al.* [KBB$^+$00]. Meshes generated from range data are usually much too dense for real-time animation, due to the high resolution of modern range scanners. While in principle mesh decimation methods can be applied to obtain a mesh resolution suitable for animation [CMS98], much better results are achieved by adapting a generic head model to the scan data, making it also possible to skip initial mesh generation from the scans. The approach presented by LEE *et al.* [LTW95] uses cylindrical Cyberware range scan data. Using the inherent planar parameterization of such scans, a flattened generic face mesh is fitted in the plane to features of the texture, obtaining the final vertex positions from the corresponding depth map. The moving facial parts eyeballs, eyelids, teeth, and tongue are modeled as separate geometric entities. MARSCHNER *et al.* [MGR00] also start out with laser range scans and construct a subdivision surface with displacement maps from that data. The subdivision surface is constructed on a generic control mesh, which is fitted to the scan data by continuous optimization over vertex positions. The final mesh is generated by subdividing the control mesh a number of times and computing displacements along the vertex normals. Noteworthy in MARSCHNER's work is also a generic eye model, which is matched to and stitched into the face mesh to achieve a seamless blend.

Volume Scans

In medical applications, volume scans are the preferred method of head geometry acquisition, allowing to obtain insights into the internal bone and tissue structure. Typically data from *computed tomography* (CT) or *magnetic resonance imaging* (MRI) is used. If surface data is additionally acquired using a range scanner, it must be registered to the volume data set [KGC$^+$96, KGPG96]. This segmentation can usually not be performed fully automatically, and is thus aided by interactive procedures, requiring a lot of experience on the side of the user. Hard and soft tissues can be separated by segmentation [DCH88]. For simulation of facial surgery, KOCH *et al.* [KGC$^+$96] connect a face mesh to the bone structure obtained from CT data. In a similar context, AOKI *et al.* [AHTN01] construct a generic head model including skin surface and skull structure from CT data. For each subject, two x-ray images and a frontal facial image are marked interactively with landmarks. The generic model is then adapted to these images.

3.3 Rendering Faces

Whatever method is used for modeling a face, finally it has to be shown on the screen. In most real-time systems, polygon meshes are used for rendering head models, enhanced with textures from photographs [LWMT99, LTW95, PHL$^+$98, GGW$^+$98]. PIGHIN *et al.* [PHL$^+$98] note that the construction of a single texture map from multiple photographic views leads to slight blurring due to resampling and registration errors. They improve rendering results by mapping and blending the input photographs in a view-dependent manner during run time. Since until very recently bump mapping was too expensive to be carried out in real time [TCRS00], subtle detail such as facial wrinkles is often simulated by additional color textures [LWMT99, WKMMT99]. For control of rendering speed versus quality, level-of-detail methods have been applied to animated faces built from polygons [SMT00].

In less time-constrained settings, the are more possibilities for improvement of skin appearance. WU *et al.* [WKMMT99] provide several rendering options, enabling bump mapping or displacement mapping for more realistic rendering of wrinkles or skin micro structure. DEBEVEC and MARSCHNER acquire the reflectance field of facial skin [DHT$^+$00, MWL$^+$99, MGR00] under controlled light. Since a photographed face is always lit according to the specific conditions under which the image is taken, much effort is put into removal of lighting information, so that a face can be rendered under new, simulated lighting. When synthetic faces are modeled, this type of capturing problems is not present, but great care still has to be taken to achieve skin rendering with natural appearance, especially in movie production [A$^+$01].

In non-real-time settings, facial wrinkles can be represented in the geometry. VIAUD *et al.* [VY92] construct explicit bulges along isoparametric lines of the spline surface used for the face model. Creases and wrinkles can also elegantly be supported when a subdivision surface model is used [DKT98]. Smooth scalar fields can also be constructed on such a surface, which is useful for parametric texture mapping and procedural shaders.

3.4 The Parameterization Problem

Given the geometric representation of a head model, the question arises how the animation should be controlled. Seen from the implementation side, the most obvious method is direct manipulation of the defining parameters of the geometry, i.e., the control points of a spline surface, the vertices of a polygon mesh [Par72], etc. This technique gives the animator maximum control, and is thus still widespread in the artistic domain. On the other hand, composing facial expressions in this manner is extremely cumbersome and requires a lot of knowledge about the possible contortions of faces. Unexperienced users are at a loss, because the configuration space (the number of possible vertex positions) is virtually infinite, but the number of *meaningful* configurations that correspond to facial expressions is very small.

This explicit pose construction can be eased by capturing expressions directly from real individuals [PHL$^+$98, BV99] and constructing new expressions by blending vertex positions in between. If a range scanner is used for capture, only face postures that can be statically held can be acquired. The range of expressions that can be generated is determined by the extremes of the captured poses, so some loss of generality has to be expected in this approach: there is no theory that confirms the existence of a base of face poses for the complete space of facial expressions.

As PARKE has noted early on [Par82, Par74], it takes much less effort to produce an animation if, instead of working on the geometry definition level, a suitable parameterization is used. Another beneficial effect of using parameterized models is improved compressibility: a small number of parameters specifies a complete expression. By transferring only these parameters instead of a complete image or facial geometry, an animated face can be quickly transmitted over a low bandwidth channel [Par82]. This idea is reflected in the facial animation part of the MPEG-4 Standard [Koe02].

3.4.1 What is a Good Parameterization?

When creating a parameterization for facial animation, a number of important aspects have to be considered:

Completeness: Do the parameters allow specification of *all* facial expressions that are possible in reality?

Restriction: Can *only* meaningful configurations be built with the parameterization?

Interpretation: Is the effect of the parameters intuitive and clear to the user?

Size: What is the minimum size of a useful parameter set?

Developing an optimal parameterization is difficult. As PARKE writes [Par82], there are two basic approaches: defining ad hoc parameters supporting direct creation of the desired deformations, and on the other hand, deriving parameters from the structure and anatomy of the face.

Eventually, the usefulness of a parameterization is limited if it is not flexible enough to form the complete range of facial expressions. According to the psychologists EKMAN and FRIESEN, there are about 55000 differentiable facial expressions, which convey roughly 30 different meanings [EF77]. The so-called *primary expressions* signal emotions: anger, disgust, fear, happiness, sadness, and surprise. EKMAN and FRIESEN have systematically collected and analyzed facial expressions, and developed the *Facial Action Coding System* (FACS). This system is based on activation of facial muscles and thus independent of cultural or personal interpretations. Expressions are modeled by 66 *Action Units* (AU), each of which describes the activation levels of one or more muscles. The AUs are grouped by muscle location into upper and lower face, and by direction into vertical, horizontal, oblique, and orbital. A third group includes miscellaneous actions such as jaw grinding, head movement, and eye rotation. The value of FACS for facial animation systems lies in the validation potential: the quality of a parameterization can be evaluated by trying to reproduce the expressions as specified by FACS.

The parameterizations found in current facial animation systems model either the visible features or the underlying muscle structure to varying degrees of realism. Both types are discussed in the following.

3.4.2 Feature-Based Parameterization

In his parameterized face model, PARKE [Par82] uses *expression parameters* to control the animation, including eyelid opening, eyebrow position, eye direction, jaw rotation, upper lip position, and mouth articulation. A different set of *conformation parameters* is used to shape the head model. These parameters include neck, chin, and cheekbone shape, eyeball size and position, jaw width, and nose length. Building on this system, COHEN and MASSARO use parameters controlling tongue length, angle, width, and thickness for realistic speech synthesis [CM93]. In both systems, the face geometry is directly distorted by adjustment of the parameters. Interestingly, each parameter specifies an interpolation value between two extreme positions of a subset of the model vertices, relating this parameterization approach to pose interpolation.

Motion of facial features can also be described by the shifting of feature points defined on the face. In the system described by GUENTER *et al.* [GGW$^+$98], the movement of a point is directly translated to deformation of the nearby mesh geometry. The MPEG-4 standard [Koe02] also employs feature points, making a distinction between *facial definition parameters* (FDP), which define the shape of a face, and *facial animation parameters* (FAP), which control animation. The FDP set is comprised of 84 standardized feature points on the face. The FAP set provides several layers of abstraction: expression and viseme parameters allow high-level specification of facial expressions. Lower-level parameters represent elementary facial motion, providing control over eyes, tongue, mouth, head orientation, and even the nose and the ears. For each animation parameter, the assignment to the influenced feature points is specified in the standard. How movement of the feature points is in turn mapped to deformations of the skin geometry is left to the application.

In general, care has to be taken by the animator to avoid unrealistic postures when using a feature-driven approach. On the other hand, such a system leaves enough freedom for cartoon-like, exaggerated motions.

3.4.3 Muscle-Based Parameterization

FACS can be seen as a muscle-based parameterization of the face since the encoded actions describe expressions by muscle activation values. Action Units provide an abstraction, which can be used to compose facial expressions. The AU notation is also independent from any particular face: actions are always relative to a subject's neutral base face. The muscle representations used in facial modeling and animation are relatively simple compared to other muscle models already used in computer graphics, for instance, the finite element muscle model by CHEN and ZELTZER [CZ92], which simulates the deformation of individual skeletal muscles. Also SCHEEPERS *et al.* [SPCM97] specify the shape of skeletal muscles by arranging ellipsoids and bicubic patches for complex muscle geometry.

For facial animation, WATERS [Wat87] presents a geometric muscle model which uses functions based on cone-shaped, rectangular, and elliptic areas to model linear, sheet, and sphincter muscles. In this model, the skin is represented as a membrane with no thickness, not reflecting the layered structure of real tissue and muscles. The geometry of a muscle and its region of influence on the skin are not separated: the skin deformation resulting from shortening of a muscle is computed directly from the corresponding contraction parameter value, within the bounds prescribed by an area of influence and a fall-off function. Combining the influence of multiple muscles on one skin vertex is difficult, and ad hoc mechanisms for resolving the conflict must be devised. Still, the method is computationally cheap and requires no numerical simulation.

In physics-based approaches, on the other hand, the skin deformation is not directly specified, but emerges from forces acting on the skin layer, applied by attached virtual muscles. Subtleties such as wrinkles and furrows emerge without explicit modeling. The first approach based on a rough physical simulation of muscles and facial anatomy appearing in the literature is due to PLATT and BADLER [PB81]. Facial actions are caused by muscle contraction on groups of abstracted simple fibers. A fiber is represented by a vector from a *muscle point* to a bone attachment. The muscle point is attached via simplified springs to nearby skin nodes. In contrast to WATERS's approach, the muscle and skin layers are conceptually separated. A muscle does not have an actual geometry, though, and the authors recognize that there is no mechanism to make the muscle follow the shape of the skull. An extension to a piecewise linear muscle model is proposed, which could be used to lay out muscles in a more flexible way. TERZOPOULOS and WATERS [TW90] use a similar muscle model to drive facial expressions. When contracted, a muscle pulls at the deeper nodes of a tri-layer tissue model. Each muscle vector has a range of influence, using a cosine blend profile to combine contributions to the movement of a skin layer node from multiple muscles. In an extension to a piecewise linear muscle fiber representation by LEE *et al.* [LTW95], where each fiber now also has a specified width, the muscles are able to follow the shape of the skull on contraction. Restricting to animation of 2D images, WATERS and FRISBIE [WF95] focus specifically on mouth animation. The muscles are represented by a simple mass-spring system, with the upper and lower halves of the orbicularis oris modeled as cross-braced bands, while all other muscles are simple springs.

Enhancing the geometric representation of facial muscles, WU *et al.* [WMTT94] use a surface-based representation. For each muscle, a set of key points is specified on the skin and then offset inwards. From these displaced points, a surface of revolution is computed. Springs from the skin nodes attach to the muscle surface. A variation of this model employs B-spline patches for representation of the muscle surface [WKMMT99], where the covered skin area can be specified interactively. The issue of the necessary muscle layout specification for each individual head model is not addressed by many authors. LEE [LTW95] and AOKI [AHTN01] include the muscle structure in their deformation process that matches a generic head model to individual head geometry.

3.5 The Interpolation Problem

When the parameterization of a face model has been chosen, parameter values for the creation of a desired facial expression can be specified. Changing these parameters over time produces the actual animation. Even with a parameterization that allows perfect modeling of all facial expressions, it is still non-trivial to synthesize an animation convincingly, because of the complex dynamic behavior of the face: a transition between facial expressions in reality is characterized by a complex choreography of moving and changing facial features. Simply interpolating all animated parameters together at the same speed will not look realistic. In a keyframing system, realistic motion thus requires laborious specification of many individual parameter values. In contrast, performance-based approaches attempt to capture the motion directly from an individual and turn it into animation parameters. An animation can also be synthesized procedurally, using heuristics that are possibly derived from empirical data.

3.5.1 Keyframed Animation

In keyframed animation, values for facial parameters are specified explicitly for important points in time, typically at extreme poses. This relates to conventional pose-to-pose animation, where the animator draws the key poses, which are then essentially interpolated by drawing in-betweens [TJ81]. Likewise, in computer animation, for each rendered frame many in-between parameter values have to be computed by interpolation. Interpolation is in the simplest case linear, easily resulting in awkward, non-smooth motion if not enough keyframes have been specified. Various spline functions are thus commonly used for interpolation of keyframes.

Keyframes can be either edited using GUI systems, or specified by scripting languages such as the SMILE system proposed by KALRA *et al.* [KMMTT91]. In SMILE, different levels of abstraction are provided that allow easy pose specification on a high level as facial expressions and phonemes, as well as direct tuning of low-level parameters. The script language also allows the combination and synchronization of visual speech and expressions.

Combining separate animations into one is a general problem with keyframing systems: in pure facial expressions, motion in one part of the face typically does not influence the other part [EF77]. For instance, movement of the eyebrows is decoupled from lip movement. This is not true, however, for animated emotional speech if the parameters for a talking mouth must be merged together with those for a happy, smiling expression: both involve mouth movements. How parameters should be combined in such a case is generally an unsolved problem, which is either avoided by ruling out conflicting cases [PWWH86], or treated ad hoc by adding up parameter values influencing the same face region [Pel91].

3.5.2 Performance-Based Animation

The shortcomings of keyframed animation are circumvented when the motion is captured directly from a real actor's face and then transferred to the virtual head model. GUENTER *et al.* [GGW$^+$98] use markers on the actor's face and define the deformation of the geometry directly from the tracked movement of these points. For parametric models, it is necessary to derive the animation parameters from the captured performance. TERZOPOULOS and WATERS analyze frontal face images and derive contraction values for their muscle-driven model [TW90]. Using a very similar face model, LUCERO *et al.* [LM99] record intramuscular EMG data from the muscles of the lower face, using hooked-wire, bipolar electrodes which are inserted into the skin. The obtained values are converted to muscle activation values for the virtual model to produce speech animations. The correlation between animation and observed real motion (captured by infrared diodes) is examined to validate the simulation results. For expressive animation, ESSA [EP97] has implemented a computer vision system to track and analyze facial expressions from video sequences. Muscle control parameter values are estimated from the optical flow and a physical model of the skin and muscle structure. The system is also used for recognition of facial expressions.

Transferring an existing animation to different face geometry can also be seen as performance-based animation: the motion of one synthetic face is tracked and mapped onto another face. NOH *et al.* [NN01] set up dense surface correspondences between the source and the target model using a semi-automatic feature point detection procedure. The transfer of the animation is then carried out on the geometry level by mapping vertex motion from one model to the other.

3.5.3 Synthesized Animation

Capturing animations directly is not always practical: an actor is needed to perform, and additional hardware may be required, at the very minimum a video camera. For every piece of animation such as a text read by a virtual news person, the scene needs to be acted out by a real person—a rather inefficient production concept. Since keyframing an animation manually is equally unattractive from the efficiency point of view, it is desirable to have techniques to *procedurally* generate new animations. Most research in this area has concentrated on production of visual speech. Commonly, a database of mouth positions (visemes) corresponding to phonemes is created, and a coarticulation model is employed to synthesize realistic lip movement for a new phoneme sequence. The visemes are often manually constructed by observation of real speakers [CM93, WL93] or automatically from digitized video input [WF95]. In video-based systems, input video is directly analyzed and used to synthesize new utterances [BCS97, Bra99, EGP02]. There are also attempts to procedurally enhance existing animation, to deliver more liveliness [LBB02], or to change the expressiveness of the animated face, for instance, adding feelings of nervousness, lightheadedness, sadness, etc. [BB02].

3.6 Skin Tissue Simulation

Deforming the face geometry based on purely geometric parameterizations can only provide a limited degree of realism. This concerns the nuances of facial expressions, such as wrinkles and furrows, as well as the dynamics of the face: the biomechanical properties of skin are reflected in the way it deforms over time when it stretches or relaxes. Physics-based systems simulate the dynamics of facial skin using Newtonian mechanics. Since the visible changes on the skin surface result from the interaction of several tissue layers, these systems necessarily need to approximate the anatomical structure of the human head. Layered construction is not only used for physics-based animation, though: CHADWICK *et al.* [CHP89] shape the skin of a character by layering bones, muscles, fat tissue, and skin into a lattice, which can be deformed by free-form deformations [SP86]. In anatomy-based modeling, virtual characters are also constructed in a layered manner, focusing on the shape of bones and muscles [SPCM97, WV97].

In the context of facial animation, the majority of approaches restricts to the simulation of the skin's elastic properties, i.e., the undeformed shape is completely restored after forces are removed. The process of wrinkle formation can only be simulated by including viscosity, though. Highly viscous materials do not store deformation energy, and thus don't return to the reference shape after unloading.

3.6.1 Mass-Spring Models

In a mass-spring model, the surface or volume of a material is approximated by mass points connected by elastic springs. The dynamics of such a system are governed by a set of differential equations: the Lagrange equations of motion. Deformation of shape is the result of applying external forces such as gravity or pull to the mass nodes. Numerical integration of the equations of motion over time describes the movement of the mass particles until the spring mesh finds a new stable configuration. Such a system is very easy to implement if simple integration schemes such as Euler's method are used. Therefore, physical simulation using mass-spring systems has become very popular in computer graphics, especially in cloth animation, where the structure of woven material lends itself to approximation as a quadrilateral mesh of connected springs [BHG92].

As on of the first to use a mass-spring approach in facial animation, the system proposed by PLATT and BADLER [PB81] models the skin as a network of springs with no thickness. Additional springs connect some of the nodes to the underlying muscles. The simulation itself approximates the behavior described by the equations of motion: a muscle force is propagated to the points attached to the muscle on the face mesh. From these forces, a displacement of the points is directly computed based on the stiffness constants of the adjacent springs. Time discretization is performed by applying a force not at once, but fractions of it over a number of steps. This is mainly done to distribute the effects of several muscles pulling simultaneously.

TERZOPOULOS and WATERS [TW90] present an advanced three-layered model of the skin, consisting of cutaneous tissue, subcutaneous fatty tissue, and a muscle layer. This lattice consists of point masses connected by springs with varying stiffness constants, modeling the elastic properties of the skin layers. To simulate non-linear elasticity, the springs representing the dermis are biphasic, i.e., a spring's stiffness constant changes according to its current length. Due to the volumetric construction, for an initial face mesh of 960 polygons approximately 6500 spring units are used. The simulation is driven by a second-order Runge-Kutta scheme as a compromise between stability, accuracy, and speed requirements. Additional volume preservation constraints model the incompressibility of the ground substance. To this end, local restoration forces are computed to minimize any deviation in volume of the elements. The layered model structure was simplified by LEE *et al.* [LTW95] to two layers: dermal-fatty and muscles. The approach also introduces skull non-penetration constraints, which prevent the skin layer from moving through the bone. The skull is approximated as an offset surface to the skin mesh. LEE reports interactive frame rates on a multiprocessor workstation. Based on TERZOPOULOS's and WATERS's work, LUCERO *et al.* [LM99] replace the ad hoc spring mesh coefficients and muscle model parameters by more well-founded values using knowledge about the actual biomechanical properties of skin and muscles. As the focus in their work is on accuracy of simulated visual speech production, considerably more computation time is devoted to the numerical integration: a three-second animation at 60 Hz takes four minutes of computation. Also focusing on speech animation, WATERS and FRISBIE [WF95] use a two-dimensional mass-spring model of the mouth.

3.6.2 Finite Element Simulation

Finite element methods (FEM) [Bat82] are traditionally used in structural analysis of materials in engineering, but are also becoming increasingly common in computer graphics applications. FEMs work with a discretization of the material geometry being simulated, for instance, a volumetric mesh of tetrahedrons. The material properties are placed in a global stiffness matrix, which relates forces on the element nodes to the resulting displacements of these nodes. This system is solved directly for a static solution. The FEM approach is commonly viewed as being more stable, more accurate, and computationally more expensive compared to spring mesh simulation techniques. The computation times for a solution typically lie in the range of minutes, making these methods unsuitable for real-time animation. They are prevalent in medical applications such as computer aided surgery, where the geometry is typically extracted from CT data and accuracy is mandatory. An idealized three-layer finite element model of facial tissue is constructed by DENG [Den88, PW96] in her work on skin excision simulation. The outer skin layer is connected via a sliding layer to the underlying muscles. PIEPER *et al.* [PRZ92] use surface scan data for plastic surgery planning. From incision lines, drawn directly on the geometry, a finite element mesh for simulation of the cut is generated. This mesh is of much coarser resolution than the face geometry, each element covering about a square inch of skin. KEEVE *et al.* [KGPG96] adapt a generic, triangular face mesh to CT data. The triangles are then projected onto the underlying skull surface, resulting in a finite element mesh comprised of six-node prism elements, which is used for simulation of bone realignment operations.

3.6.3 Hybrid Schemes

A few researchers combine mass-spring models and finite elements. For bone repositioning applications, KOCH *et al.* [KGC$^+$96] represent the facial surface by two-dimensional non-linear finite elements, which are connected to the skull by nodal springs. The spring stiffnesses are computed from CT data. In a similar fashion, WU *et al.* [WMTT94, WKMMT99] develop a skin model that combines plasticity, viscosity, and elasticity for simulation of expressive wrinkle formation and skin aging using a skin mesh. The nodes of this mesh are connected by springs to the surface of an underlying muscle layer.

3.7 Summary

Despite the deceptively photo-realistic appearance of image- and video-based facial animations, only systems using three-dimensional geometric head models show the flexibility to enable a wide range of applications. Within this class, there is a trend towards increasingly realistic representation of anatomic detail, including parameterization by muscles and physics-based skin tissue simulation. While the former provides intuitive controls for animation, the latter can produce realistic facial deformation including subtle effects without explicitly modeling them. Some simplifications have to be made for the sake of performance, though. While there are some fairly elaborate layered models of facial skin, representations of musculature and skull are still simplistic, especially in systems geared towards interactive performance. Assembling a head model based on human anatomy takes a lot of effort, and only a few researchers report on methods for construction of such a model for the purpose of adaptation to scans of an individual. The anatomy-based head model proposed in this dissertation provides detailed approximations of the skull and the facial musculature while still being efficient enough for real-time animation. Acknowledging the importance of easy construction of such a structured head model, a general method for adaptation of a generic head template to various targets is demonstrated.

CHAPTER 4

Anatomy of the Human Head

> *No man should marry before he has studied anatomy*
> *and dissected the body of a woman.*
> *– Honore de Balzac*

The foundation of any physics-based facial animation system is a good understanding of the human head's anatomical structure. A thorough look at real faces confirms the reasonable assumption that the appearance of the face is mainly determined by the major structures of the head, namely the skull, the muscles, and the skin. If a realistic look is the goal of an animation system, none of these components can be left out from the model. The first step in building a virtual head model is thus to study the biological structures of the head and the interactions between them. In this chapter, I have collected relevant material from a variety of sources in medicine [Ras78, KLP76, Ste79, Sob01], forensics [Tay01], and computer graphics [TW90, CZ92]. After examination of the facial skin, the skull as the primary supportive structure of the head is introduced. Finally, the facial musculature is discussed, which is situated between skin and skull and responsible for driving facial expression. Although not all the detail presented here is reflected in the implemented facial animation system, it is still instructive to get an impression of the fascinating complexity of the human head. Inevitably, a few medical terms must be used to discuss anatomy. As far as these are not explained directly in the text, they can be found in Appendix A.

4.1 Facial Skin Structure

The human skin is a complex system, with different components working together to serve a variety of functions. These include protection against external environmental influences such as bacteria or toxic substances, the regulation of body temperature, and storage of water, fat, and blood. Each square centimeter of skin contains on average 1 meter of vessels, 15 sebaceous glands, 100 sweat glands, 3000 nerve ends, and 3,000,000 cells. The skin is the largest sensory organ, receptive for sensations of touch, temperature, and pain. Skin also plays a vital role in human communication by means of facial expression, texture, color, and scent.

As shown in Figure 4.1, the skin is structured into three layers: *epidermis*, *dermis*, and *hypodermis*. Together, the topmost layers epidermis and dermis are called the *cutis*, the underlying hypodermis is also known as the *subcutis*. In the follow-

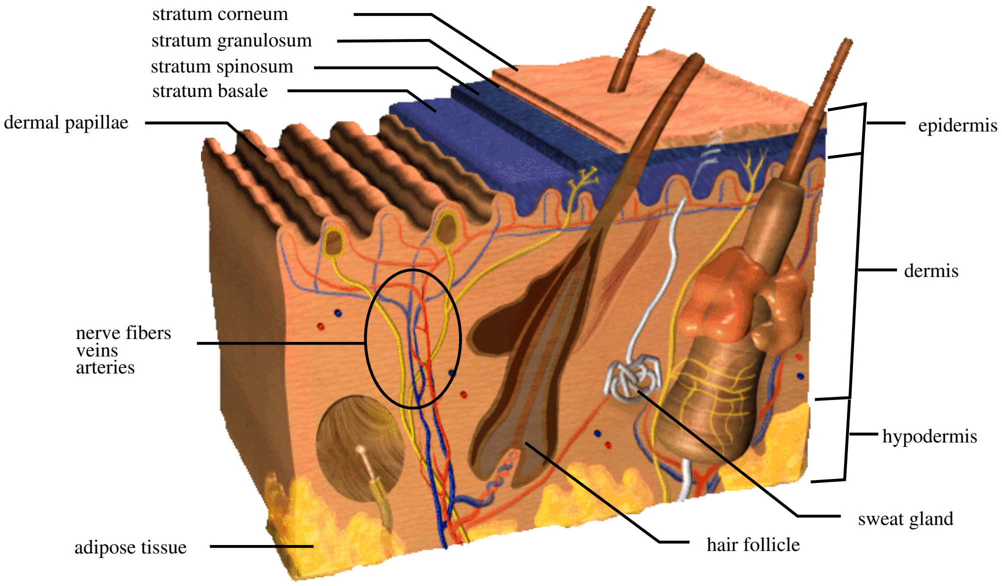


Figure 4.1: Cross section of human skin. Image: [BW97]

ing the layers of the skin are briefly examined, focusing on their basic structure. The biomechanical properties of skin are discussed in Chapter 6.

4.1.1 Epidermis

In the face, this superficial layer of the skin is about 0.02 mm thick, which is about 3–10% of the total skin thickness. The epidermis itself has no vessels or glands. It is organized into three functionally different zones: cells are born in the *stratum basale*, differentiated and matured in the *stratum spinosum* and the *stratum granulosum*, and finally layered and disposed of in the *stratum corneum*. The life of a cell lasts about 30 days. The stratum corneum consists mostly of the protein keratin, and gives protection against acids and alkali. It is capable of binding water, giving this layer smoothness and elasticity. The stratum corneum is rather loose on the surface, but becomes more compact in its deeper parts. In the stratum basale, the melanocytes are found, cells that produce the pigment melanin, which protects the skin from ultraviolet rays. A thin layer between the stratum corneum and the stratum granulosum, the *stratum lucidum*, plays an important role for the optic appearance of skin: it is highly refractive, creating an optically homogeneous layer.

4.1.2 Dermis

The dermis is divided into the *stratum papillare*, a layer that consists of tiny papillae projecting into the epidermis, and the *stratum reticulare*. The structure of the stratum papillare differs regionally with the mechanical demands onto the respective part of the body surface: at the eyelids, for instance, the papillae are less developed than at the elbows. This layer consists of loose connective tissue with capillaries, elastic fibers, reticular fibers, and some collagen. The stratum reticulare provides strong resistance to tearing forces. It is made from dense connective tissue, containing large blood vessels, nerve endings, hair follicles, and other cells. The intercellular substance is comprised mostly of collagen fibers and the amorphous, gel-like ground substance. The collagen fibers are arranged in thick wavy bundles, layered parallel to the surface. They uncurl when the skin is stretched, resulting in variable resistance of the skin to forces.

4.1.3 Hypodermis

The hypodermis serves as a fat storage and isolator. Fat cells are layered between septa of collagenous, connective tissue. The fat increases the tension of the skin, and isolates the body against thermal influences. The development of the subcutaneous fat depends on sex, age, body region, mechanical demands, hormonal influences, nutrition, and other factors. In the face, some parts such as the eyelids and the external ear are fat-free. The hypodermis is connected to and glides on the fascia, a layer of tissue that connects the muscle surface to the skin.

4.2 The Skull

The fundamental bone structure of the head is the skull, which gives the head stability and its basic shape. The bone material is not a dead substance, but a living tissue that adapts to the forces induced by the functional movements of the facial musculature over a lifetime. Also, growth of the brain and of the surrounding braincase influence each other. In anatomical terms, the skull is divided into the braincase (*neurocranium*) and the face (*viscerocranium*). Each part consists of many separate bones, which are shown in Figure 4.2.

4.2.1 Neurocranium

The forehead and the upper portion of the orbital cavity surrounding the eyes are formed by the *frontal bone* (*os frontale*). The *supraorbital ridge* (*arcus superciliares*) is the prominent part underlying the eyebrows and is usually more pronounced in males. The *occipital bone* (*os occipitale*) forms most of the base and back of the neurocranium. The brain is connected to the spinal cord through a hole in this bone. The other bones of the neurocranium come in pairs, mirrored at the median plane: the *parietal bone* (*os parietale*) makes up part of the top and the side of the cranium. The widest part of the head is usually formed by the protrusion of this bone behind the ear. The *temporal bone* (*os temporale*) and parts of the *sphenoid bone* (*os sphenoidale*) shape the lower side of the neurocranium. Attached to the *mastoid process* (*processus mastoideus)* of the temporal bone is the sternocleidomastoid muscle, which flexes the neck. Other features of the temporal bone include the ear hole (*meatus acusticus externus*) and a zygomatic process named *zygoma* (*arcus zygomaticus*) as part of the cheek bone.

4.2.2 Viscerocranium

Itself part of the viscerocranium, the paired *zygomatic bone* (*os zygomaticum*) connects to the zygoma on the neurocranium, thus completing the *zygomatic arch*. Other bones of the viscerocranium that are mirrored on either side of the skull are the *maxilla*, the

	neurocranium
1	frontal bone
2	supraorbital ridge
3	occipital bone
4	parietal bone
5	temporal bone
6	sphenoid bone
7	mastoid process
8	ear hole
9	zygoma
	viscerocranium
10	zygomatic bone
11	maxilla
12	nasal bone
13	lacrimal bone
14	ethmoid bone
15	mandible
16	ramus
17	temporomandibular joint

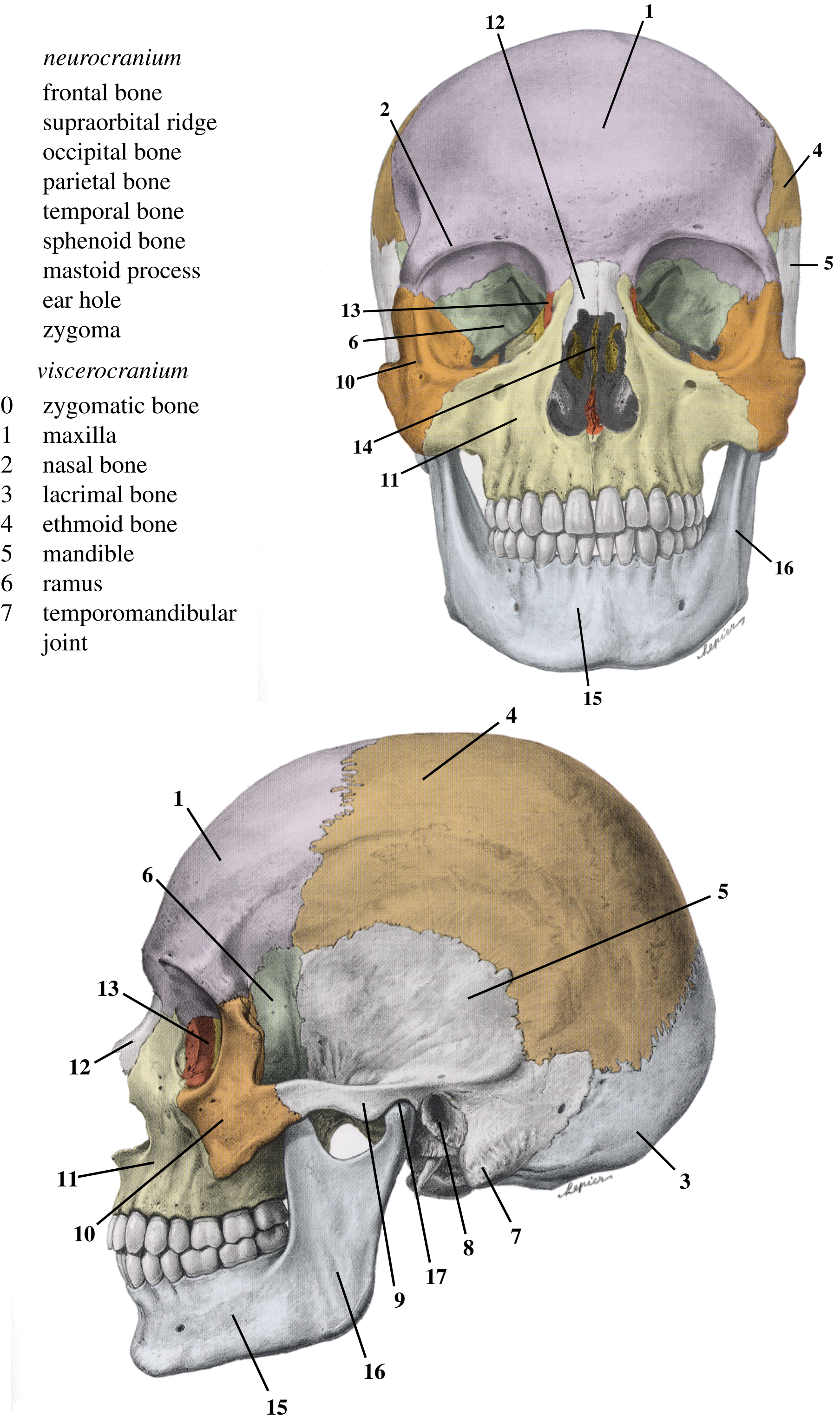


Figure 4.2: Human skull, front and side views. Images: [Sob01]

nasal bone (*os nasale*), and the *lacrimal bone* (*os lacrimale*). The maxilla forms the upper jaw and, together with the zygomatic and lacrimal bones, shapes the lower part of the orbital cavity. The nasal bone gives shape to the upper part of the nose at the beginning of the nasal bridge. Unpaired bones of the middle face are the *palatine bone* (*os palatinum*) at the root of the mouth and the *ethmoid bone* (*os ethmoidale*) in the nose.

The *mandible* (*mandibula*) completes the viscerocranium. It is separated from the rest of the adult skull and hinged to the neurocranium at the *temporomandibular joint*. The degrees of freedom of this joint allow for opening and closing of the mouth as well as chewing motion. Functionally, the temporomandibular joint is a composition of two joints: when the mouth is opened, a rotation in the lower part of the joint is superposed on a horizontal translation to the front in the upper part. The posterior, upright angle of the mandible is called the *ramus*. The *hyoid bone* (*os hyoidale*), which is situated at the root of the tongue (see Section 4.4.2), is also counted as a part of the skull. It is connected to the base of the skull only by ligaments.

4.3 Facial Muscles

The richness of facial motion is made possible by the numerous muscles lying between skin and skull. The muscles of the face are grouped into two categories: the *muscles of mastication* and the *muscles of expression*. While the former are responsible for chewing and grinding food, the latter enable facial play. Some muscles have a role in both groups, though. More than 50 muscles can be easily identified on the head, and, at a closer look, many muscles can be considered groups of smaller muscles themselves. Therefore, the number of facial muscles given in the literature varies. There are also many deeper muscles attached to various parts of the head, which are not anatomically considered muscles of the face, and are also for the most part not mentioned here. The following section briefly discusses the general structure and mode of action of muscle tissue. In Sections 4.3.2 and 4.3.3, the muscles of the face are then presented from a mainly functional point of view.

4.3.1 Muscle Physiology

Physiologically, facial muscles belong to the *skeletal muscles* (the other muscle types are *smooth* and *heart* muscle). The components of a single muscle are shown in Figure 4.3. A muscle cell has cylindrical shape, is up to 15 cm long, and 10–100 μm wide. Each cell contains numerous cylindrical structures 1–2 μm wide, called *myofibrils*, which run over the whole length of the cell. The myofibrils in turn consist of a repeating pattern of *sarcomeres*. This pattern results in the banded appearance of some skeletal muscles under the microscope, alternating light and dark—hence the name *striated muscle*. The sarcomeres are the functional units of the muscle, having the ability to contract upon excitation. The combined effort of many sarcomeres across the muscle leads to its contraction and causes visible deformation of the skin through the connective tissue. Most muscles in the face originate at the bone structure, while their surface inserts into the skin. When contracting, a muscle thus pulls the skin towards its fixed origin. In contrast, some facial muscles attach to bone on both ends or only to the skin.

There are two major modes of muscle contraction: *isotonic* and *isometric*. Under isotonic contraction the load to be lifted is fixed, allowing the muscle to shorten at a maximum speed limited by the load. The force the muscle generates is constant. On the other hand, isometric contraction occurs if the muscle is not allowed to shorten any further while it is contracting, for instance, if the load is too heavy to lift.

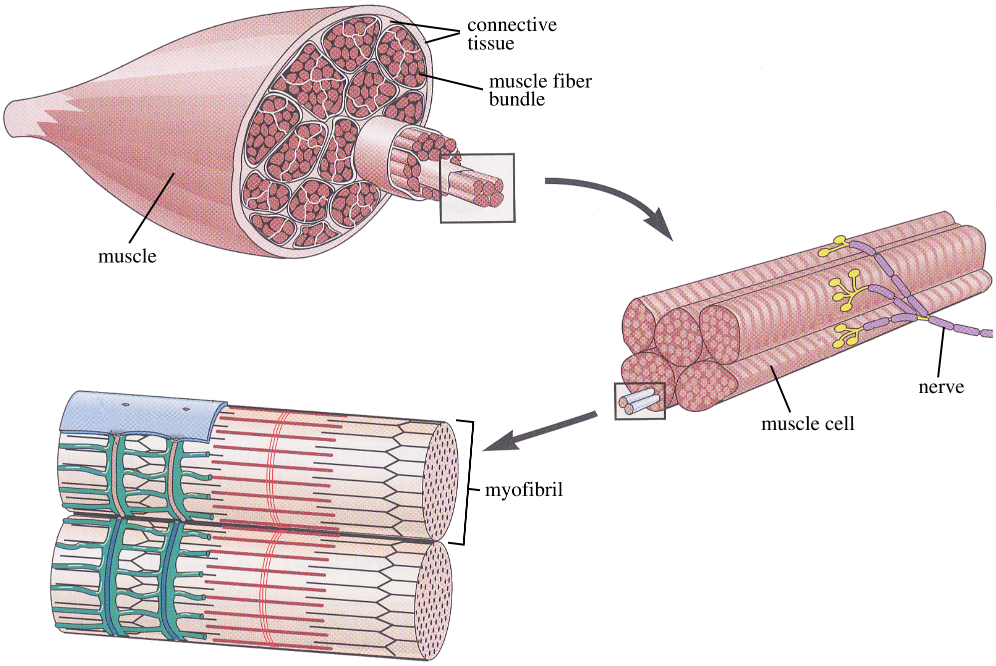


Figure 4.3: Structure of a muscle fiber. Image: [Spe98]

4.3.2 Muscles of Expression

The muscles of expression insert into the facial skin, causing distortions of the skin under contraction. When skin is pushed together through muscle contraction, this often causes wrinkling, where the direction of the wrinkles is always perpendicular to the direction of the muscle's line of action. In the anatomy literature, the muscles of expression are categorized by their region of influence: the scalp, the eyelid, the nose, the mouth, and the neck. Figures 4.4 and 4.5 show frontal and side views of the facial musculature. Most muscles in the face are mirrored on either side. Where the following muscle descriptions use the singular, equivalent behavior for the corresponding muscles on both sides is implied.

Scalp

Together, the muscles of the scalp form the *epicranius*. This broad layer, which covers the whole of one side of the vertex of the skull, is very tightly attached to the skin. Its two major parts are the *occipitalis* at the back of the head and the *frontalis* covering the forehead, together named *occipitofrontalis*. This pair of muscle bellies is connected by a tendinous sheet, the *epicranial aponeurosis*. The epicranial aponeurosis is also the origin of the *temporoparietalis* muscle. The occipitalis originates in the skull behind the ears, while the frontalis is not attached to bone, but inserts into the skin of the forehead and the eyebrow region, originating approximately at the hairline. The temporoparietalis extends from the epicranial aponeurosis to the top of the ear on the side of the head. The epicranius moves the skin of the head. Notably, the frontalis lifts the eyebrows and eyelids, causing also wrinkles on the forehead. Next to the temporoparietalis, the three parts of the *auricularis* attach to the auricle and the spine of the helix. Their function is to move the ear.

Eyelid

The circular muscle running around the eye is named *orbicularis oculi*. It consists of the thick *pars orbitalis*, encircling the orbit, and the *pars palpebralis*, which underlies the upper and lower eyelids. At the inner corner of the eye, the circular bundles of fibers are interrupted by the *palpebral ligament*. The pars orbitalis attaches to the inner orbit and the skin of the cheek. When contracting in a pursing manner, this muscle supports tight closure of the eyelids. The group of its medial fibers in the upper eyelid region that fans out into the eyebrow is also called *depressor supercilii*. The tender pars palpebralis attaches to the cartilaginous *tarsal plate* (*tarsus*) in the eyelid (see Figure 4.6), and causes blinking on contraction. The look of the eye is influenced by the tarsal plates when the orbicularis oculi contracts. Also, the contraction forms the small radial wrinkles at the outer corners of the eye known as crow's feet. To open the eye, the *levator palpebrae* raises and folds back the upper eyelid. This muscle is not regularly counted as a facial muscle, but belongs to the group of *extra-ocular muscles* (see Section 4.4.1). It originates deep inside the bony orbit of the eye and attaches to the upper eyelid. The *corrugator supercilii* penetrates the orbicularis oculi and the frontalis. It pulls the corners of the eyebrows together and downward, creating a vertical crease in the skin above the nose.

Nose

The *procerus* muscle covers the lower part of the nasal bone and the upper part of the lateral nasal cartilage. It inserts into the forehead between the eyebrows. This muscle pulls down vertically, causing a horizontal crease over the bridge of the nose. The *nasalis* covers the nose wing. It inserts into the side of the nose, above the canine tooth and the incisor. Contraction of the nasalis pulls the nose wing down and back, closing the nostril. The *levator labii superioris alaeque nasi* originates at the inner side of the eye orbit. It inserts into the skin of the upper lip and the nose wing, which is lifted on contraction, thus widening the nostril. Strong contraction of this muscle induces a furrow to the side of the nose. The *levator labii superioris* originates from the maxilla under the eye orbit and lifts the lateral part of the upper lip.

Mouth

The muscle encircling the mouth is the *orbicularis oris*. It covers the area from the bottom of the nose to halfway down the chin. Though it primarily acts in a sphincter-like fashion to close the mouth, this muscle is not a simple sphincter. It has superficial and deeper layers of muscles, which act independently. It also consists partly of fibers

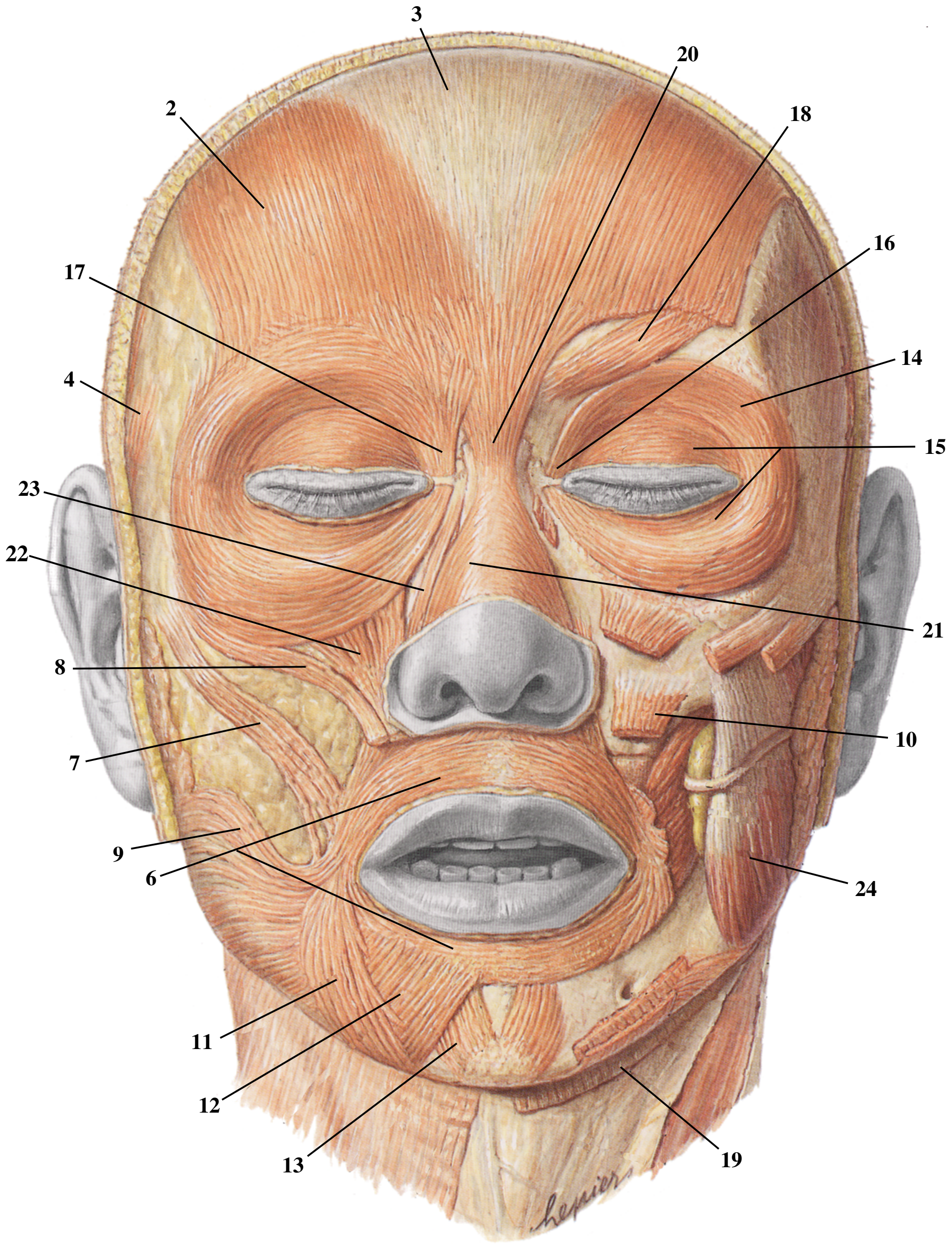
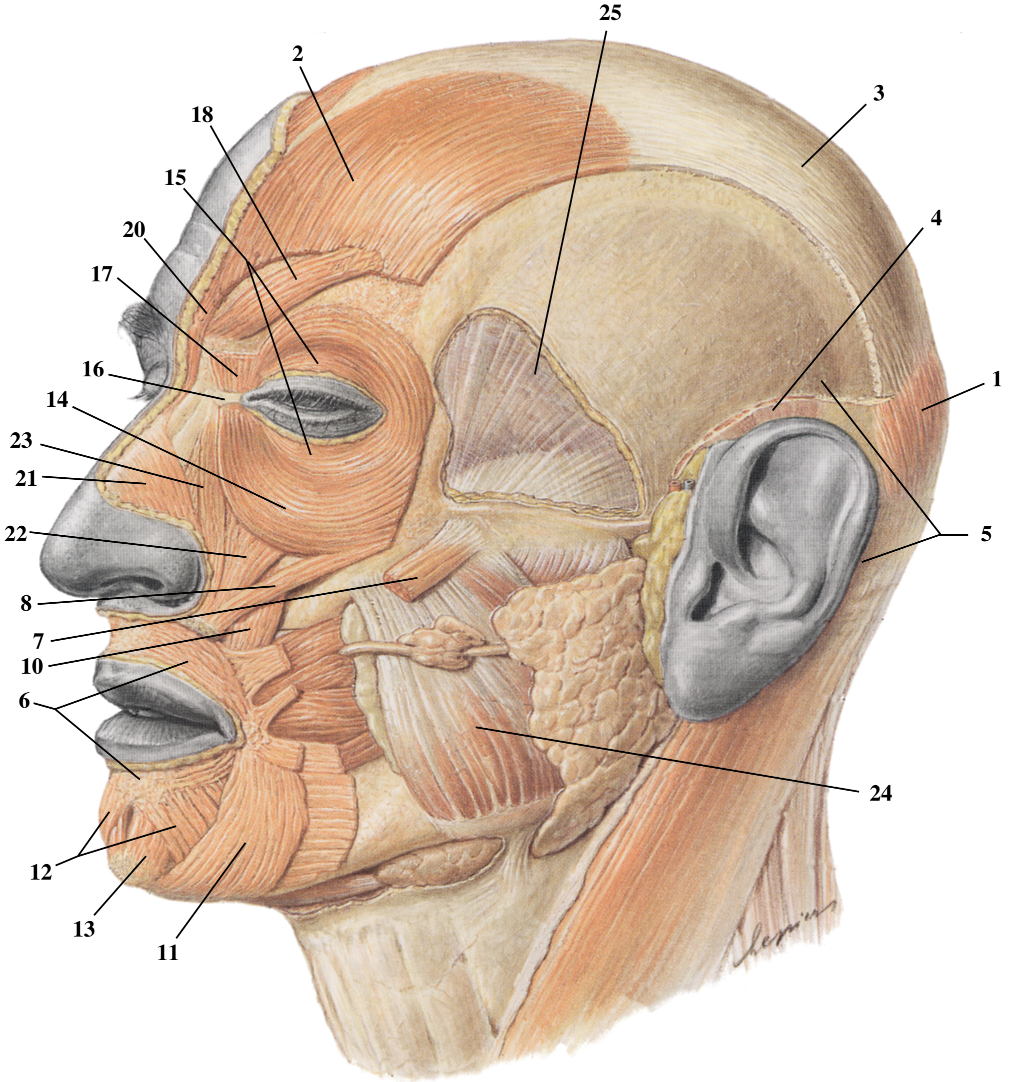


Figure 4.4: Facial muscles, front view. Image: [Sob01]



	scalp		*mouth*		*eyelid*
1	occipitalis	6	orbicularis oris	14	o. oculi, p. orbitalis
2	frontalis	7	zygomaticus major	15	o. oculi, p. palpebralis
3	epicranial aponeurosis	8	zygomaticus minor	16	palpebral ligament
4	temporoparietalis	9	risorius	17	depressor supercilii
5	auricularis	10	levator anguli oris	18	corrugator supercilii
		11	triangularis		
		12	depressor labii inferioris		
		13	mentalis		
	neck		*nose*		*m. of mastication*
19	platysma	20	procerus	24	masseter
		21	nasalis	25	temporalis
		22	levator labii superioris		
		23	l. labii sup. alaeque nasi		

Figure 4.5: Facial muscles, side view. Image: [Sob01]

from other facial muscles, which insert into the lip region. The orbicularis oris thus forms an intricate structure that allows for the rich variety of facial motion pertaining to the mouth. This includes protrusion and retraction of the lips as well as a major role in speech articulation.

The *zygomaticus major* and *zygomaticus minor* muscles run from the zygomatic bone to the mouth corners, blending into the upper lip part of the orbicularis oris. They help creating a smile by lifting the upper lip near the mouth corner. The *risorius* pulls the corner of the mouth back horizontally. It does not originate from the bone, but in the fascia of the masseter muscle (see Section 4.3.3). The *levator anguli oris* inserts into the orbicularis oris at the corner of the mouth and aids smiling by pulling upwards. Its fibers intermingle with the zygomaticus, the orbicularis oris and the *triangularis*. The triangularis inserts into the mouth angle and pulls it downwards. The lower lip is pulled down by the *depressor labii inferioris*. It inserts into the orbicularis oris and the skin below the lower lip, originating at the lower edge of the mandible. The *buccinator* runs horizontally along the cheek to the corner of the mouth. It originates at the inside of the mandible, with three attachments there. This muscle aids in chewing by keeping the cheeks close to the teeth, thus keeping the food on the tooth surface. It is also used for sucking action, and sometimes causes a dimple in the cheek to appear.

The chin is covered by the *mentalis*. This muscle originates just below the teeth, and inserts into the skin at the ball of the chin. By pulling upwards, this muscle raises and protrudes the lower lip. If the separation between the pair of mentales muscles is large enough, this leads to a visible cleft on the chin.

Neck

From the more than 20 muscles extending down the neck, only the *platysma* is part of the facial musculature in the anatomical classification. It originates at the base of the mandible, and covers the bottom of the jaw completely, extending down to the front part of the neck. Functionally, it stretches the skin of the neck.

4.3.3 Muscles of Mastication

The *masseter* is a very strong muscle that runs diagonally across the cheek. Its origin lies at the zygomatic arch, and it inserts into the jaw angle. When contracted, this muscle tightly clenches the teeth, thus playing the major part in grinding food. The *temporalis* also aids in closing the mouth: it is a broad, radiating muscle, covering the side of the head. Originating at the temporal bone, the muscle goes through the zygomatic arch and inserts into the coronoid process of the mandible. The *medial pterygoideus* or *internal pterygoid* muscle is located on the side of the jaw and assists the temporalis and masseter in raising the mandible to clench the teeth. In contrast, the *lateral pterygoideus* or *external pterygoid* muscle helps to open the mouth, and to protrude the mandible. Asymmetric action of these muscles on either side of the head result in side-to-side movements of the jaw, as in chewing action.

4.4 Assorted Organs

The visual appearance of the face is, of course, not determined by skin, muscles, and skull alone. Other vital components of the human head are the eyes and the oral cavity with teeth and tongue. These are briefly described in the following.

4.4.1 Eyes

The eyes are the visual sensory organs. Each eyeball (*bulbus oculi*) is located in an orbit of the neurocranium, as shown in Figure 4.6. The eyeball is about 24 mm in diameter, being roughly sphere-shaped with a bulge at the *cornea* on the front side. six of the seven *extra-ocular muscles* enable rotation of the eyeballs in its socket. The seventh extra-ocular muscle is the levator palpebrae superioris, which lifts the eyelid (see Section 4.3.2). The iris has two muscles controlling the size of the pupil, the *dilator pupillae* and the *sphincter pupillae*. The pupil diameter can change between 2 mm and 8 mm, accommodating for the external lighting conditions. The eyelid provides protection against strong light and intrusion of foreign particles. It consists mainly of the tarsal plate, covered by the orbicularis oculi muscle and a thin skin layer. The lid keeps the cornea moist through blinking, distributing the lacrimal fluid (tears) secreted from the *lacrimal gland*. The lacrimal gland is situated on the lateral side of the upper eyelid and contributes to the well-rounded appearance of the skin there. The inside of the eyelid skin is connected to the eyeball, thus forming a bag of connective tissue, which is open to the *palpebral fissure* at the front. The sebum secreted from glands in the tarsal plate gives the fissure its typical specular appearance. A substantial influence on the appearance of the eyes is also due to the brows and eyelashes.

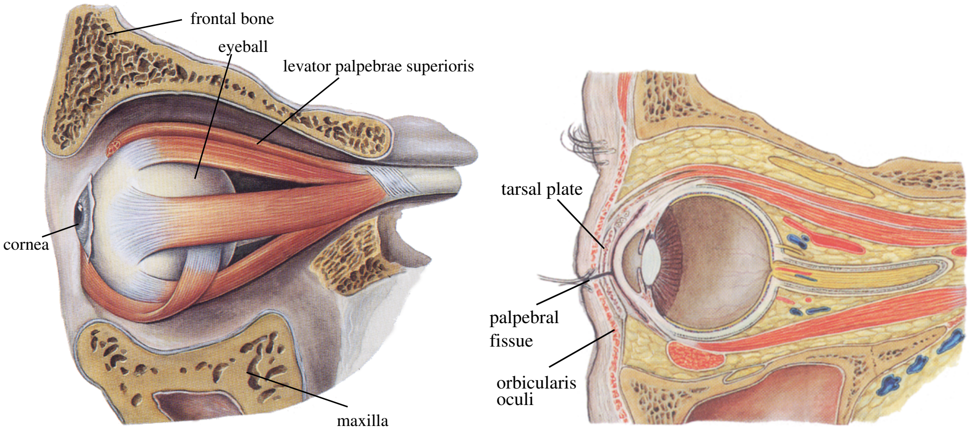


Figure 4.6: Eyeball in its socket and in cross section. Images: [Sob01, Spe98]

4.4.2 Mouth

The *oral cavity* (*cavitas oris*) is the enclosed region between the cheeks and the lips. Figure 4.7 shows a view into the oral cavity. The major part of the actual cavity can only be seen when the mouth is wide open. The roof of the mouth is built from the

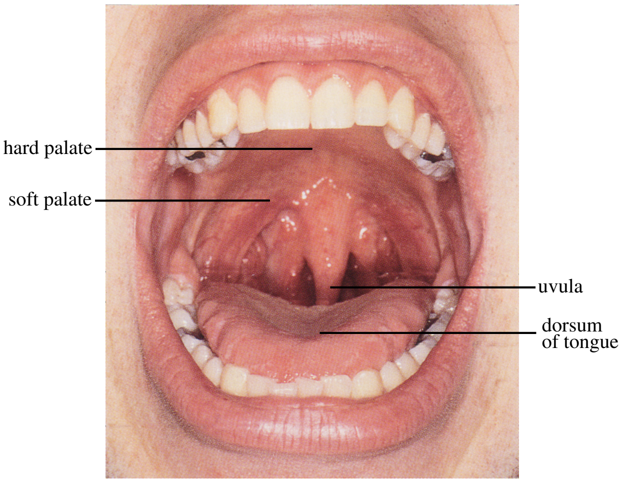


Figure 4.7: A view into the oral cavity. Image: [Sob01]

hard palate in the front, and the *soft palate* in the back. The *uvula* is attached to the soft palate.

The oral cavity is lined by the teeth of the upper and lower jaw. The teeth help cutting and grinding food. They are attached to the upper and lower jaw bone by long roots. Usually only the enamel-covered top part of a tooth, the *crown*, is visible. The transition to the root is the *neck* of the tooth and is covered by the gums. The full adult set is comprised of a total of 32 teeth, 8 teeth per jaw half: 2 incisors, 1 canine tooth, 2 premolar and 3 molar teeth. The lower jaw with teeth is shown in Figure 4.8.

When the mouth is closed, it is mostly filled by the tongue (also see Figure 4.8). The *dorsum* of the tongue, its top surface, is composed of a mucous membrane, covered with papillae which are sensitive to the four main tastes. The body of the tongue is mainly comprised of muscular tissue. Eight muscles provide for the flexibility of the tongue's movements. They connect it to the mandible, the hyoid bone, and the temporal bone. The tongue shapes the food in the mouth, shifts it onto the teeth, and transports it towards the throat. It is also vital for speech articulation.

A number of saliva glands below the skin create some well-rounded areas in the face region around the mouth: the *parotid gland* surrounds the jaw angles, and the *submandibular gland* is situated along the lower edge of the mandible.

4.5 Discussion

This introduction into the anatomy of the head is by no means exhaustive, but already shows an immense richness of detail that is hard to capture in a simulation. Furthermore, the current medical knowledge is mostly qualitative, and it is difficult to obtain exact quantitative measurements. There is a significant degree of uncertainty about the anatomy of the facial musculature [LM99]. For exact modeling of muscle dynamics, for instance, precise data about muscle lengths and cross-sectional area during contraction would be needed. Also, tissue properties often differ from one individual to the next, for instance, thickness and elasticity of skin. Many of these properties can-

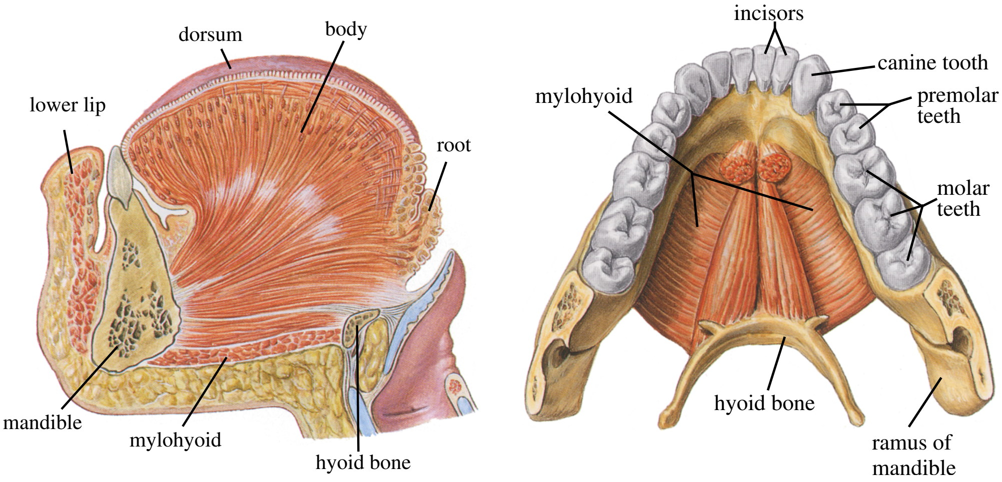


Figure 4.8: Left: the tongue in cross section; right: mandible with teeth and hyoid bone. Images: [Sob01]

not be easily obtained by non-intrusive measurements. Still, while a truly complete and accurate reconstruction of the head is currently an insurmountable challenge, it is nevertheless possible to build a working model that incorporates many insights gained from medical research and which delivers high quality facial animation. Compromises have to be made for the sake of computational efficiency, but the fundamental approach of simulating real anatomy allows continuous incorporation of advances in medical knowledge to improve the model. Additionally, progress in numerical simulation techniques and faster computers will enable the use of more precise approximations to real anatomy.

CHAPTER 5

System Overview

> *All parts should go together without forcing. You must remember that the parts you are reassembling were disassembled by you. Therefore, if you can't get them together again, there must be a reason. By all means, do not use a hammer.*
>
> *– IBM maintenance manual, 1925*

The integration of anatomical details of the human head into a facial animation system is a non-trivial task. As the model becomes more and more complex, it also becomes increasingly difficult to assemble and adjust all the parts to arrive at a consistent, animatable model. Furthermore, modification of the geometry of the virtual head requires a rebuild of the internal structure. To simplify the construction of head models, I propose the use of a generic template that incorporates all the structural elements needed for animation. This template is the basis for construction of individualized, animatable head models, which are created by deformation of the prototype, adapting it to given target constraints.

Other than in the typical artistic pipeline, where geometry is first modeled and then equipped with controls for animation, the head model is immediately animatable in my approach. With each deformation, not only the outer skin geometry is deformed, but also the internal structure composed of muscles and skull. As a beneficial side effect, all head models derived from the reference head share the same set of animation parameters, enabling re-use of animation scripts.

In this chapter, I describe the structure and construction of the reference head model. In preparation of the following chapters, an overview of the components of the complete facial modeling and animation system is given.

5.1 The Reference Head Model

The template head model, as shown in Figure 5.1, encapsulates six major structural components:

- a triangle mesh serving as the **skin surface**. The mesh edges are aligned to facial features to reduce animation artifacts, and the tessellation is adjusted to the deformability of the facial regions. The algorithms working on the mesh frequently use adjacency relations such as finding the triangles sharing a given edge, etc. To this end, the mesh data structure is based on a half-edge representation [CKS98].

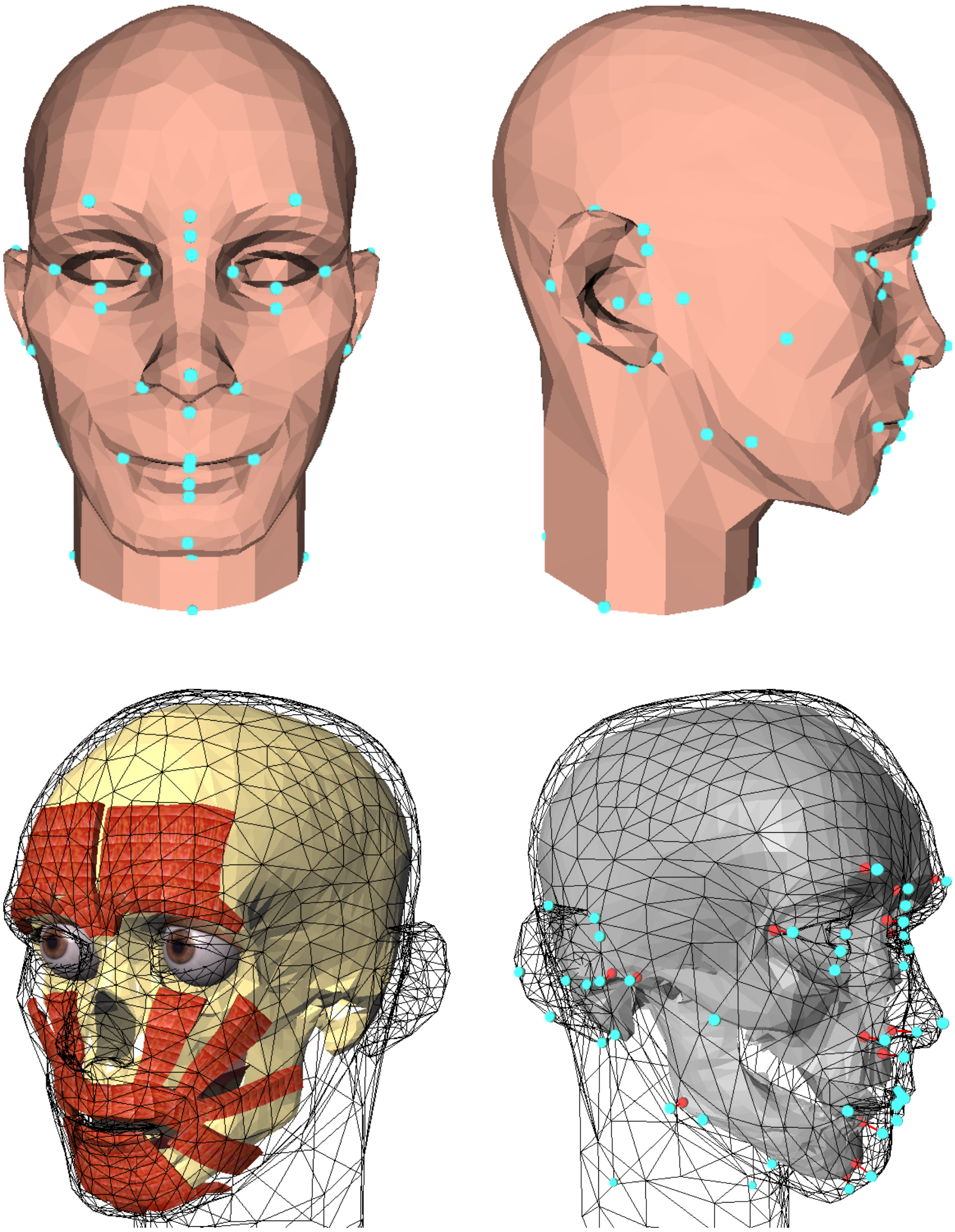
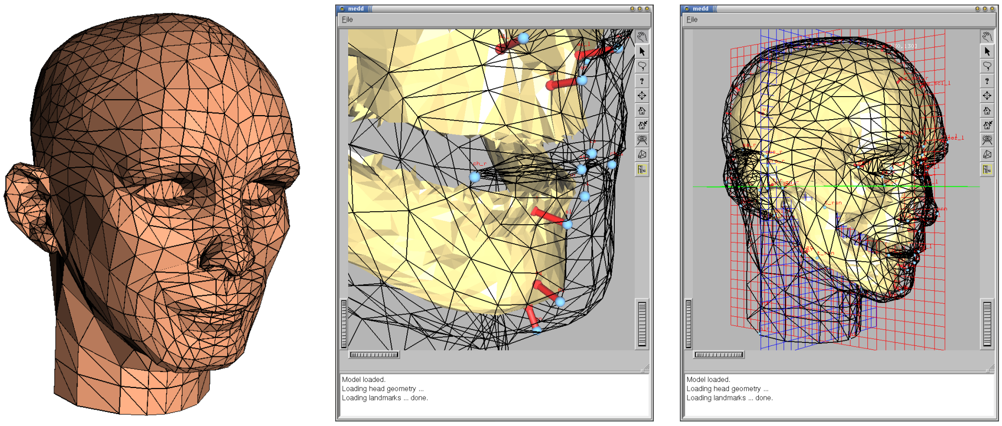


Figure 5.1: Reference head geometry and structure. Top: head geometry with landmarks (blue dots), front and side views; bottom left: skull and facial components; bottom right: skull landmarks (red dots) related to subset of skin landmarks.

- 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. The current system includes 24 of the major muscles responsible for facial expressions and speech articulation. The muscle model is discussed in Chapter 7.
- an embedded **skull**, including a rotatable mandible, to which skin and muscles attach. The skull model is also represented as a triangle mesh, but the geometry is only used for initialization of the structure, not during the actual animation.
- a **mass-spring system** connecting skin, muscles, and skull. 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. Properties and construction of the spring mesh are discussed in Chapter 6.
- separately modeled geometric objects for **eyes, teeth, and tongue**.
- a set of **landmarks**, defined on the skin and skull surfaces. I use these landmarks to control deformation of the head structure. The landmarks follow the conventions laid out in the anthropometric literature [Far94], where a minimum subset of landmarks is chosen according to their prominence in the face and existence of correspondences between skin and skull. How the landmarks are used to define the deformation of the head model is described in detail in Chapter 8.

5.1.1 Reference Head Construction Process

The generic head model is built manually by designing a face mesh first and then putting in the anatomical structure. As discussed in Section 3.2.2, it is extremely difficult to obtain a suitable animatable face mesh directly from 3D scans [HKA$^+$01]. The outer geometry of the reference head is thus constructed using a commercial 3D modeling package. For the very regular structure required in the face region, the simple face model used by PARKE [PW96, Appendix B] provides a good starting point.



```
Model loaded.
Loading head geometry ...
Loading landmarks ... done.
```

```
Model loaded.
Loading head geometry ...
Loading landmarks ... done.
```

Figure 5.2: Interactive construction of the reference head model. From left to right: reference mesh; landmark editor closeup showing skin and skull landmarks; skull matched to skin mesh in the editor.

For the non-animated part of the head—outside the face region—a simplified mesh obtained from a range scan of a real person is stitched to the generic face. The resulting head mesh as shown in Figure 5.2 consists of 1027 vertices and 2030 triangles. This mesh resolution is a compromise that delivers satisfying visual quality, while the complexity is kept low enough to allow the physics simulation to run in real time.

A triangle mesh representing an anatomically correct human skull is used as the base for the skull component, using a freely available model. The skull model consists of two pieces, separating the jaw from the remainder of the skull. The skull and jaw meshes consist of 2062 and 2066 triangles, respectively. As mentioned above, the complexity of these meshes does not influence the animation at run-time: the geometry is only used to compute the distance between skin and bone and to automatically attach the lower parts of the face to the mandible, so that skin and musculature follow the jaw rotation.

Using a custom editing tool, landmarks are placed on the skull and skin surfaces, as shown in Figure 5.2. To match the generic skull model to the reference head geometry, the skull is deformed using the methods described in Chapter 8. The head model is also aligned to a standard posture, the *Frankfort horizontal* (see Section 8.2), so that landmark distances on the model correspond to distance measurements listed in the anthropometric literature. This is important for the applications demonstrated in Sections 8.5 and 8.6.

The design of the facial muscle structure is based on medical literature on human anatomy. The muscles are interactively laid out on the head model employing the editing methods I discuss in Chapter 7. Finally, separate geometric objects for the facial components eyes, teeth, and tongue are added, resulting in the final model shown in Figure 5.1. These components consist of procedurally generated polygonal geometry, depicted individually in Figure 5.3.

Given all parts, the mass-spring system that connects skin, muscles, and skull is computed automatically to obtain the fully animatable reference model. Figure 5.4 shows the spring mesh in the mouth region of the head model.

The construction approach described here is in many ways complementary to recent techniques that model articulated human or animal body models from the underlying anatomy [WV97, SWV02]: a creature is developed from the inside out, starting with the skeleton, attaching muscles, and finally layering the skin on top. In contrast, in the approach presented here the underlying anatomical structure is built to match

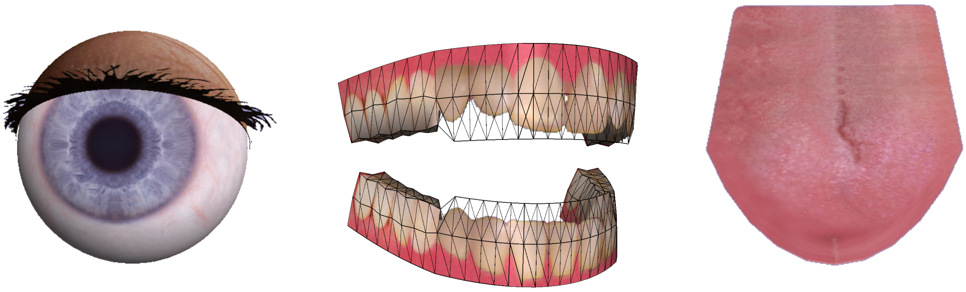


Figure 5.3: Eye, teeth, and tongue models

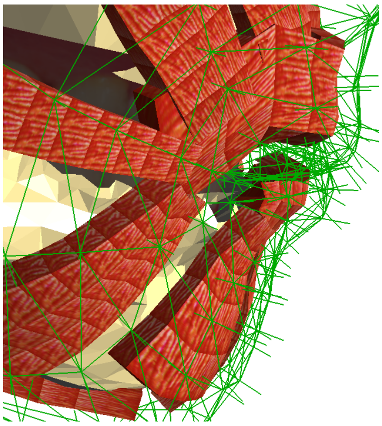


Figure 5.4: Closeup of mouth area with muscles, skin, and skull connected by springs

the already known outer geometry of the face. Once the labor-intensive assemblage of the parts is completed for the template model, individual head models can be created from it with ease.

5.2 Physics-Based Simulation

The head model is mainly animated by contracting the virtual muscles, causing deformation of the facial skin. Shortening a muscle results in stretching of the springs connecting the muscle to the skin, thus inflicting forces onto the mass nodes. A physics-based simulation, based on the Lagrangian equations of motion, computes the resulting displacements of the particles, until their movement leads to a new equilibrium, i.e., a stable state of the system. The particular spring mesh structure, discussed in detail in Chapter 6, is geared towards real-time animation. By the structure of the spring system, volume preservation and skull non-penetration constraints are integrated into the global solution of the equations of motion, obviating the need for local geometric criteria [LTW95].

5.3 Animation Control

Facial animation is not only produced by specification of muscle contraction values, but also from the additional parameters head rotation and position, eyeball rotation, eyelid movement, tongue rotation and translation, and rotation of the jaw. The values for all these parameters are generated in several ways:

Expression editing: In the expression editor (see Figure 5.5), the animation parameters can be changed while the simulation is running, resulting in an immediate change of facial expression. Complete parameter snapshots can be saved as expressions and later be reused for animation by sequencing in time. Blends between expressions are achieved by linear or spline interpolation between parameter values for each frame. While this editing method provides direct access to all degrees of freedom of the face, the complex dynamics of facial motion are hard to reproduce in this manner.

eyelid_right 0.1862087101
head_rot_x -1.799999952
head_rot_y -9
head_rot_z 0
head_trans_x 0
head_trans_y 0
head_trans_z 0
jaw 0
look_at_x 240000
look_at_y -50000
look_at_z 1000000
tongue_open 0

```
Reading muscle data from
/HPS/usr/kaehler/projects/medusa/examples/kolja/kolja-31-bigeyes
(using affine head model transform)... done.
```

Figure 5.5: Expression Editor. Left: current expression shown in the running simulation; top right: adjustable muscle parameters; bottom right: expression palette.

Speech from audio: ALBRECHT uses the animation system to generate synthetic visual speech from audio input data, employing an algorithm that transforms phonemes into visible mouth shapes (visemes) [Alb01, AHS02b]. For realistic speech animation, coarticulation effects have to be taken into account. To some extent, additional expressive animation can be generated from the audio data to make the speech more lively [AHS02a].

Speech from text: Current text-to-speech synthesis modules perform extensive linguistic analysis of the input text to produce a speech signal. Instead of working from audio input, this phonetic transcription can be used to generate speech animation [AHK$^+$02]. By including available information about pauses and the types of sentences, as well as interpreting additional text markup in the form of emoticons (“smileys”), emotional expressions can be supported.

Face tracking: Some of the facial parameters are tracked using computer vision techniques, as shown in Figure 5.6. This results in very natural motion, but not all subtleties of the face can be captured easily. In the current system, only eye blinks, head rotation, and head translation are recorded. Still, the result of the abovementioned speech animation techniques is vastly improved by using these recorded parameters in addition to synthesized mouth articulation. The input consists in this case of video footage, with audio, of a speaking person.

5.4 Deformation of the Reference Head Model

To obtain individual head models, the reference head is deformed using a technique based on the landmarks attached to skin and skull. The landmarks serve as handles for controlling the shape of the head and the facial features. Using such anthropometrically meaningful controls for deformation enables modeling operations on a higher level of abstraction, for instance, changes of facial proportions. The general deformation method proposed in Chapter 8 supports a multitude of applications, three of which

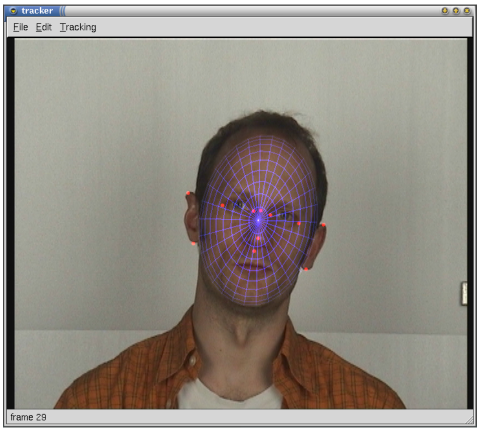


Figure 5.6: Face tracking tool. Using a number of feature points (red dots), position and orientation of the head are tracked (blue ellipsoid).

are explored in this dissertation: matching the generic model to individual range scans, simulating human head growth, and reconstructing a face from skull data. Figure 5.7 exemplifies the creation of a structured head model from scan data and later modification by changing the age of the model. As can be seen from the figure, the head model remains fully animatable throughout these transformations.

5.5 Rendering

The face mesh and the facial components are rendered as textured polygons using OpenGL [WND96]. For a head model resembling a real individual, the textures for skin, eyes, and teeth are composited from a number of photographs of that person, as described by TARINI *et al.* [TYHS02]. Figure 5.8 shows a skin texture generated in this manner and the resulting head model. Modern graphics hardware can easily handle hundreds of thousands of textured triangles per second, so highly detailed geometry can in principle be rendered in real time. Nonetheless, the numerical simulation that

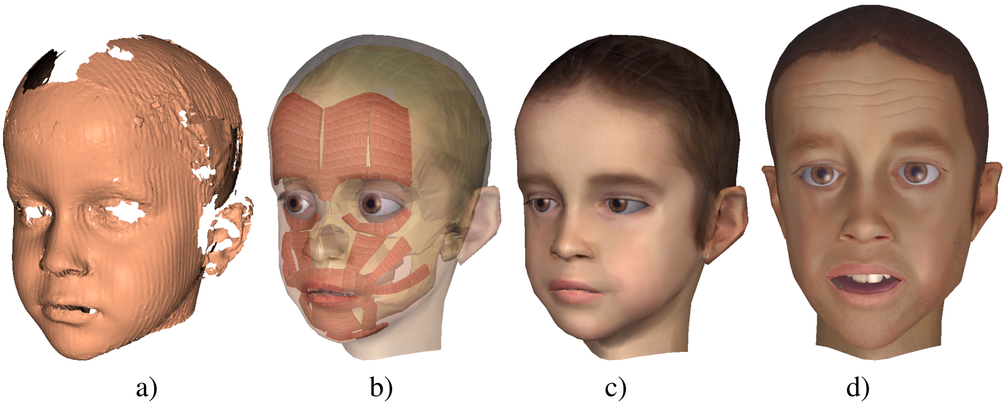

a) b) c) d)

Figure 5.7: Head models generated from range scan of a 5 year old boy: a) scan data; b) adapted head structure; c) textured; d) age changed to 20 years, surprised expression.

Figure 5.8: Visualization of the head model. Left: texture generated from photographs; right: model rendered using OpenGL.

drives the animation of the face mesh is restricted to rather coarse meshes to reach sufficient update rates. To improve the visual quality of the rendered images, different strategies are used: geometric detail can be pretended by shading tricks, or the rendered geometry is generated by refinement of the animated one using subdivision techniques. Chapter 9 discusses this in detail.

5.6 Multithreading

Fast simulation updates in the range of 30–40 fps have to be maintained for highly dynamic types of animation, for instance, visual speech. On a single-CPU system, the majority of the computational load is on the numerical integration, leaving little resources for rendering. Also, by visualizing the current simulation state in lock step, the rendering frame rate is bound to the speed of the simulation. With dual processor machines becoming more commonplace, it is thus advantageous to let simulation and rendering threads run on separate CPUs. The rendering thread can generate interpolated in-between frames for smoother animation and perform higher quality shading while waiting for the next simulation result [HKA+01]. A schematic overview of the setup for speech animation in a multithreading environment is depicted in Figure 5.9.

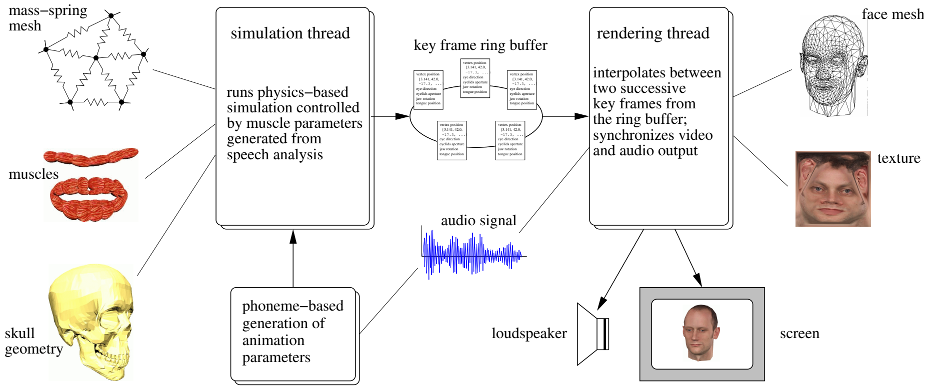


Figure 5.9: Multithreaded simulation and rendering in a speech animation framework

CHAPTER 6

Skin Tissue Dynamics

> *All the beauty of the world, 't is but skin deep.*
> *– Ralph Venning*

The excursion into anatomy in Chapter 4 has shown that the structure of facial skin is highly complex, making proper simulation difficult. In this chapter I take a closer look at the mechanical properties of facial skin and describe how simulation of skin dynamics is integrated into the facial animation system. Section 6.1 thus introduces the foundations of the biomechanical properties of skin. Other than in medical applications, where accurate simulation of real skin behavior is absolutely essential, the focus in my real time animation system is on typical and plausible behavior: skin deformation should appear realistic to the eye, but it is not mandatory that the skin on the virtual head moves and folds in precisely the same way as on its real counterpart. Adhering to this guideline, a fast mass-spring approach (see Section 6.2) is used for the simulation, employing the explicit numerical integration scheme discussed in Section 6.3. Apart from the numerical simulation, the structure of the spring mesh is vital for behavior and run-time efficiency of the virtual skin. This structure is presented in Section 6.4. The chapter concludes with a discussion of the overall simulation approach and performance measurements.

6.1 Biomechanics of Facial Skin

The examination and understanding of the mechanics of living skin is in the realm of *biomechanics*, where the mechanical behavior of biological materials under various conditions is analyzed [Fun93]. Topics in this discipline are stress and strain, strength, creep, fracture, heat transfer, vibrations, and many more. In the context of a facial animation system, results from biomechanics give insights into the deformation of facial skin under the influence of external forces such as muscle contractions and jaw movement. In a typical biomechanical experiment, forces applied to a material are measured in relation to the size of the specimen. This relation is expressed through the *stress* $\sigma = f/a$, i.e., the force per unit cross-sectional area. Its basic unit is *newton per square meter* $(\mathrm{N}/\mathrm{m}^2)$. The deformation of a solid, resulting from stress, is described by the *strain measure* e. For something as simple as a one-dimensional spring of rest length l_r that is stretched to length l, the change in length can be expressed by many different strain measures, for instance,

$$e_1 = \frac{l - l_r}{l_r} \qquad \text{or} \qquad e_2 = \frac{l^2 - l_r^2}{2l_r^2},$$

behaving similarly at low strains, but putting a different emphasis on larger strains. Deformation can take many forms besides uniaxial stretching, such as bend or shear. The usefulness of a strain measure is determined by the stress-strain relationship: for instance, a spring that is subject to a pull, stretches. Thus, an expressive graph of the applied stress over the strain e_1 can be plotted. Pertaining to uniaxial stretching of many so-called *Hookean materials* such as steel, the simple formula

$$\sigma = Ee \tag{6.1}$$

holds true for a certain range of stresses. E is a constant of the material, called *Young's modulus*, and is measured in the same unit as the stress.

In general, most real-world materials exhibit much more complex behavior. The properties of a material are described by its *constitutive equation*. The Hookean elastic solid is one of three idealized materials with simple equations, the other two being the non-viscous fluid and the Newtonian viscous fluid. As Equation (6.1) shows, a Hookean elastic solid exhibits a linearly proportional relation between stress and strain. The strain measure e of this constitutive equation can be greatly simplified by the assumption of isotropic behavior, i.e., the stress-strain relation is considered to be independent of the direction. In the one-dimensional case, the material has only one elastic constant, Young's modulus E. If another dimension is added to the solid, i.e., width, the *Poisson ratio* ν is included, describing the ratio of extension in the direction perpendicular to the line of compression when the material is squeezed.

While this and the other two idealized constitutive equations sufficiently characterize many materials used in engineering, skin and most other biological solids cannot be properly described by them. Skin has experimentally shown to have a highly non-linear stress-strain curve: under low stress, skin has low resistance against deformation, because the collagen fibers unroll and stretch. The elastin base of the skin behaves very similarly to an ideal rubber material with linear stress-strain response. With stronger deformation the collagen fibers become completely stretched and aligned with

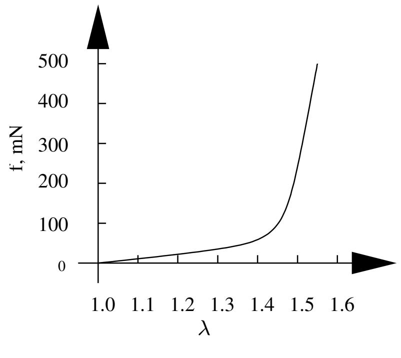


Figure 6.1: Force f vs. uniaxial stretch ratio λ for rabbit abdominal skin. Image after FUNG [Fun93].

the deformation, leading to a sharp increase in resistance. The resulting stress-strain curve, shown in Figure 6.1, is thus essentially biphasic.

Across the face, the fibers of the collagen lattice follow certain contours, the so-called *Langer's lines*. Skin deforms anisotropically along these lines, further complicating the behavior of the material. Given the resting tension of the skin, the Langer's lines indicate, for instance, the direction along which a surgical cut should be performed. Facial wrinkles also usually align to the Langer's lines.

Skin exhibits significant viscoelastic properties, i.e., it behaves partly like an elastic solid and partly like a viscous fluid. In cyclic loading, the stress-strain relationship in the loading process is thus slightly different from that in the unloading process; this effect is called *hysteresis*. Due to *stress relaxation*, the induced stresses will decrease over time, when a constant strain is applied to skin. If, in contrast, the stress is kept constant, the skin will continue to deform, a phenomenon called *creep*. These behaviors can be attributed to the incompressibility of the water in the fat cells and in the ground substance, and to the delayed motion of the fibers in the ground substance, caused by the fluids. Incompressibility also causes the *Poisson effect*: when an incompressible material is compressed in one direction, it will extend perpendicularly to the line of compression, maintaining its volume.

As can be seen from this brief introduction, realistic simulation of the complex properties of skin is a difficult task. Even given the current state of compute power available on a desktop PC, an accurate simulation using advanced FEM algorithms is by far too time-consuming for real-time facial animation. Therefore, I use spring meshes, also called mass-spring systems, a physical model that allows at least the simulation of non-linear elasticity with little computational effort.

6.2 Spring Meshes

In computer graphics, spring meshes are often used for modeling skin, cloth, and other soft materials. The deformable body is approximated by a collection of point masses, which are connected by Hookean spring elements. The resulting mesh typically describes a surface, such as a textile sheet [BHG92], but volumetric lattices can also be built [TW90]. Deformation is caused by application of forces to the mass nodes of the system. The resulting node displacements induce internal strain by stretching the connecting springs, thereby forcing other nodes to move. This process continues until the system reaches a new equilibrium, where the forces inflicted on each node cancel each other out.

6.2.1 Mass-Spring System Formulation

Formally, the point masses in such a system are represented as a set of N nodes $\mathcal{M} = \{1, \ldots, N\}$. Each node $i \in \mathcal{M}$ has a mass $m_i \in \mathbb{R}^+$, as well as associated position $\mathbf{x}_i$, velocity $\mathbf{v}_i$, and acceleration $\mathbf{a}_i$, all in $\mathbb{R}^3$. At any instant in time, position, velocity and acceleration are related by

$$\begin{aligned} \mathbf{v}_i &= \dot{\mathbf{x}}_i, && \text{and} \\ \mathbf{a}_i &= \dot{\mathbf{v}}_i = \ddot{\mathbf{x}}_i. \end{aligned}$$

The nodes are connected by a set of M springs $\mathcal{S} = \{1, \ldots, M\}$, where each spring $s \in \mathcal{S}$ connects the nodes p_s and q_s. Each spring has a current length $l_s = \|\mathbf{x}_{p_s} - \mathbf{x}_{q_s}\|$, and a rest length r_s, usually given by the spring length in the relaxed state of the system. The directional force spring s exerts on its end node p_s is then computed as

$$\mathbf{f}_s = k_s \frac{l_s - r_s}{l_s} (\mathbf{x}_{q_s} - \mathbf{x}_{p_s}),$$

where the stiffness constant $k_s \in \mathbb{R}$ determines the elasticity of the spring: the higher the coefficient, the stronger the resulting force. The spring force is a non-linear function of $\mathbf{x}_{p_s}$ and $\mathbf{x}_{q_s}$, because sums of squares are used to compute l_s. The total force $\mathbf{n}_i$ on node i is the sum of all the forces due to the springs starting *from* and the opposing forces due to the springs connecting *to* it:

$$\mathbf{n}_i = \sum_{\substack{s \in \mathcal{S} \\ p_s = i}} \mathbf{f}_s - \sum_{\substack{s \in \mathcal{S} \\ q_s = i}} \mathbf{f}_s \tag{6.2}$$

6.2.2 The Equations of Motion

The time-dependent behavior of the point masses in the mass-spring system follows *Hamilton's Principle of Least Action*, which states that the trajectory $\mathbf{x}_i(t)$ minimizes the action, i.e., the difference between the kinetic and potential energy, integrated over time. Minimizing this functional means to look for the point where its derivative vanishes:

$$\frac{d}{d\mathbf{x}_i} \left[\int_{t_0}^{t_1} L(t, \mathbf{x}_i, \dot{\mathbf{x}}_i) dt \right] = 0.$$

$L = T - U$ is the *Lagrangian*, with T being the kinetic energy, and U the potential energy of the system. The well-known Euler-Lagrange equation of motion is a consequence of Hamilton's principle, derived using specific instances of the energy terms T and U (which are not spelled out here for brevity):

$$m_i \ddot{\mathbf{x}}_i + \gamma \dot{\mathbf{x}}_i - \mathbf{n}_i - \mathbf{e}_i = \mathbf{0}, \tag{6.3}$$

where the spring force $\mathbf{n}_i$ is defined as in Equation (6.2), and $\mathbf{e}_i$ represents external forces, such as muscle-induced forces, applied to the nodes. The damping term, scaled with γ, is velocity-dependent and used to dissipate kinetic energy from the system, modeling energy loss due to friction. This is often set to a high value to stabilize the system in numerical integration, resulting in overdamped behavior.

The force resulting from a particle's mass, current acceleration, and velocity equals the sum of spring forces and external forces on node i at all times. Given the initial state of the system, the solution to the equation of motion thus describes the dynamics of the system over time, maintaining the balance between external and internal forces.

6.3 Numerical Simulation of Spring Mesh Dynamics

Equation (6.3) is a second-order ordinary differential equation (ODE). Initial positions $\mathbf{x}_i^0$ and velocities $\mathbf{v}_i^0$ are provided for all nodes i. For animation, the development of these parameters over a series of discrete time steps is needed. To solve this initial value problem, the equation needs to be integrated through time.

6.3.1 Euler Integration

A very simple and popular [LTW95, WF95] method is Euler integration, which is iterated over constant time steps Δt:

$$\begin{aligned}
\mathbf{a}_i^t &= \frac{1}{m_i}(\mathbf{n}_i^t + \mathbf{e}_i^t - \gamma \mathbf{v}_i^t) \\
\mathbf{v}_i^{t+\Delta t} &= \mathbf{v}_i^t + \Delta t \mathbf{a}_i^t \\
\mathbf{x}_i^{t+\Delta t} &= \mathbf{x}_i^t + \Delta t \mathbf{v}_i^t
\end{aligned}$$

Here, the second-order ODE has been rewritten as a system of two coupled equations of first order. Unfortunately, this method suffers from lack of stability, i.e. the time steps need to be very small if the stiffness of the system is high.

6.3.2 Verlet Integration

A method that has been designed for direct integration of second-order problems is the *Størmer-Verlet* method, introduced by the physicist VERLET in 1967 [Ver67], but also attributed to STØRMER, who used a similar method already in 1905 [Ves94]. In this formulation, the second differential quotient is replaced by the Stirling approximation

$$\mathbf{a}_i^t = \ddot{\mathbf{x}}_i^t = \frac{1}{\Delta t^2}\left[\mathbf{x}_i^{t+\Delta t} - 2\mathbf{x}_i^t + \mathbf{x}_i^{t-\Delta t}\right] + O(\Delta t^2).$$

Solving for $\mathbf{x}_i^{t+\Delta t}$ leads to

$$\mathbf{x}_i^{t+\Delta t} = 2\mathbf{x}_i^t - \mathbf{x}_i^{t-\Delta t} + \Delta t^2 \mathbf{a}_i^t + O(\Delta t^4),$$

thus computing the new node position at time $t + \Delta t$ directly from the known acceleration $\mathbf{a}_i^t$ at time t, with an error in the order of $O(\Delta t^4)$. This acceleration in turn depends on the velocity $\mathbf{v}_i^t = \dot{\mathbf{x}}_i^t$ due to the damping term of Equation (6.3). In a popular modification of the Verlet method, known as the *leapfrog Verlet* formulation, $\mathbf{v}_i^t$ is therefore estimated from the velocities $\mathbf{v}_i^{t-1.5\Delta t}$ and $\mathbf{v}_i^{t-0.5\Delta t}$:

$$\begin{aligned}
\mathbf{v}_i^t &= \frac{3}{2}\mathbf{v}_i^{t-0.5\Delta t} - \frac{1}{2}\mathbf{v}_i^{t-1.5\Delta t} \\
\mathbf{v}_i^{t+0.5\Delta t} &= \mathbf{v}_i^{t-0.5\Delta t} + \Delta t \mathbf{a}_i^t
\end{aligned}$$

Figure 6.2 illustrates the staggered evaluation order, to which the leapfrog variant owes its name. This method is not self-starting, i.e., values for $\mathbf{x}_i^{-\Delta t}$ and $\mathbf{v}_i^{-0.5\Delta t}$ need to be given to evaluate positions and velocities at $t = 0$. The first time step thus has to be bridged using a self-starting technique. Leapfrog Verlet integration is more stable than Euler integration, at comparable computation costs: the external and internal forces have to be computed only once per time step. Due to these advantages, I use the leapfrog Verlet method for simulation of the facial spring mesh, where the initial time step is performed using Euler integration.

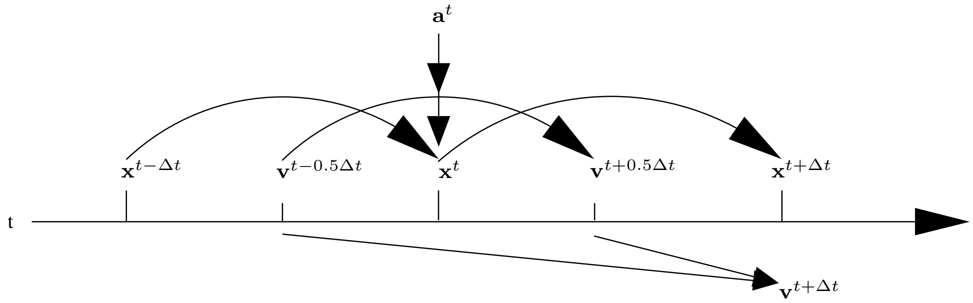


Figure 6.2: Evaluation of positions and velocities in the leapfrog Verlet method. The known acceleration $\mathbf{a}^t$ at point t is used to compute the displacement $\mathbf{x}^t \rightarrow \mathbf{x}^{t+\Delta t}$, and the velocity $\mathbf{v}^{t-0.5\Delta t} \rightarrow \mathbf{v}^{t+0.5\Delta t}$. The velocity $\mathbf{v}^{t+\Delta t}$ is approximated by averaging, and used for the velocity-dependent damping factor in the computation of $\mathbf{a}^{t+\Delta t}$.

6.3.3 Load-Adaptive Time Stepping

To maximize stability in the real-time setting, the time stepping of the simulation is adapted to the available computation time. The given, fixed frame rate r (in frames per second) determines the time slot $t = 1/r$ between two rendered frames. During the animation, the time t_x spent external to the simulation procedure is measured, mainly including updates of the muscle configuration and actual rendering. The time spent inside the simulation procedure is determined only by the time t_s needed to perform computations for all spring mesh nodes. This time is also measured. The time $t - t_x$ is thus available for the skin dynamics simulation, and $I = \lfloor (t - t_x)/t_s \rfloor$ iterations can be safely performed, each with a simulated time step of t/I. Figure 6.3 illustrates this correlation between rendering time and simulation time for one frame.

This scheme assumes that each iteration of the numerical integration algorithm takes the same time, which is a valid assumption given the constant number of nodes and springs. Overall system load can vary in a multi-tasking environment, though. Therefore, the time measurements are updated and averaged per frame to achieve better adaptation to the current computing capacity.

6.4 Spring Mesh Construction

The triangle mesh surface of the head model is used as the starting point for modeling the skin structure: direct conversion of the nodes and edges of this mesh to mass points and springs results in a spring mesh representing epidermis and dermis. The initial stiffness constant of a spring s is computed according to VAN GELDER [Van98]:

$$k_s = E_s(l_s) \frac{\sum_{t \in \mathcal{T}_s} \operatorname{area}(t)}{r_s^2},$$

where the numerator is the summed area of the (one or two) triangles $\mathcal{T}_s$ incident upon spring s, r_s is the spring's rest length, and l_s is the spring's current length, where initially $l_s = r_s$. E_s represents the two-dimensional Young's modulus of the simulated membrane. The value of E_s depends on the current length of spring s: assigning a constant stiffness to the springs restricts the simulation to linearly elastic materials,

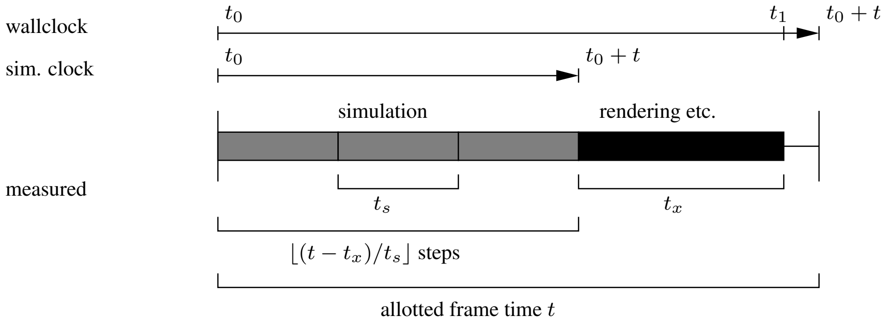


Figure 6.3: Simulation and rendering within one frame, with a prescribed time t per frame. Within that real-time interval, the simulation is advanced to $t_0 + t$ and rendering and muscle updates are performed. Measurements of computation times for one simulation step t_s and "everything else" t_x are continuously updated. Typically, some amount of time $< t_s$ is left when one frame is done, so the simulation loop is re-entered already at t_1 for processing of the next frame.

i.e., the springs always elongate proportionally to the applied strain. Thus, to model the biphasic stress-strain behavior of skin, the stiffness constant is modified at run time according to the current deformation of spring s. The values are empirically chosen as

$$E_s(l_s) := \begin{cases} 100 & \text{, if } \frac{l_s}{r_s} < 1.5, \\ 1000 & \text{, else.} \end{cases}$$

To compute the mass of the spring mesh nodes, the mass density of the skin tissue is approximated with an average value of $1100\ \mathrm{kg/m^3}$ [Duc90]. The surface area of the head model is the total area of all triangles in the mesh. Together with the assumed average thickness of the skin, the total mass of the skin is computed. This mass is finally evenly divided among the mesh nodes. For the reference head model (see Section 5.1) with 2 mm skin thickness and 1027 vertices, this results in 0.31 g per node. To avoid unnecessary computations for the static regions of the head model, the spring mesh is only constructed on the face region of the head model. For the reference head model the surface spring mesh is thus comprised of only 1791 springs and 625 nodes.

The connection to the bone structure and the muscle layer is built by inserting additional springs. To this end, static mass nodes are placed on the skull and on the muscle surface as shown in Figure 6.4(a) and connected with nodes on the surface. These muscle layer attachments are discussed in more detail in Section 7.4.6. The additional springs have rather low stiffness constants to simulate the fatty subcutaneous layer that allows skin to slide freely on the skull and on the muscle layer. The constants have again been chosen empirically: the skin-muscle connection has a stiffness of 10, and the skin-bone connection is assigned a value of 30.

The skin model as described so far has no notion of volume preservation. The skin may thus penetrate the muscle and bone layers when stretched, or even fold over. The problem is solved by attaching another spring to each mesh node that pulls the node *outwards*. The additional spring mirrors the spring that attaches it to the bone or muscle layer, also shown in Figure 6.4(a). This results in a surface node moving preferably

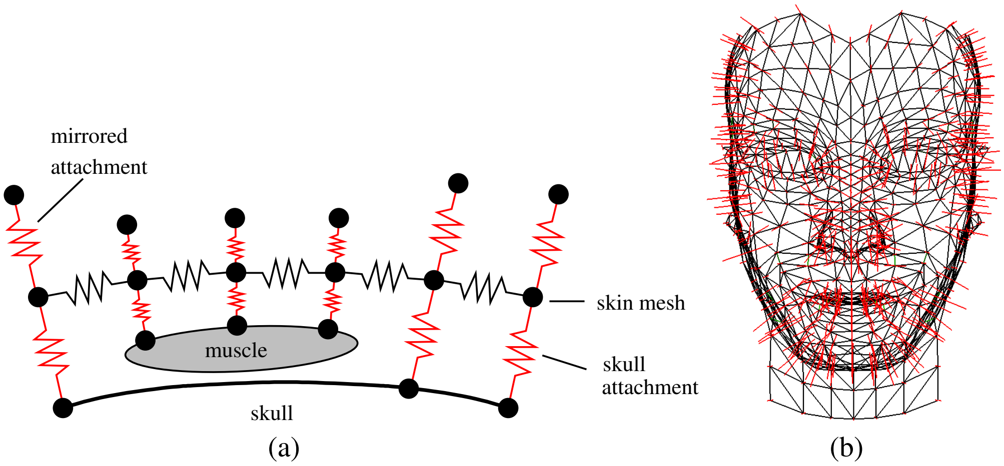


Figure 6.4: Mass-spring system. a) Springs connect the skin surface to skull and muscles; attachments are mirrored outwards. b) Springs constructed from reference head geometry. Black biphasic springs correspond to surface edges; red springs connect to bones and muscles and support skin tension.

tangentially to the skull and muscle surface. Intersection of the skin with the deeper layers is effectively avoided with this mechanism, although not completely ruled out—violent skin distortion could still cause penetration. As a physical interpretation, the mirrored springs can be interpreted as a model of the outward-directed force resulting from the internal pressure of a skin cell. Figure 6.4(b) shows the complete spring mesh for the reference head.

6.5 Discussion

The Verlet leapfrog method used here is a popular numerical integration scheme in physics [Ves94], and is also used in computer graphics applications [TG98]. A vast number of other numerical integration schemes is available, with varying advantages and disadvantages. Implicit methods currently experience a renaissance, especially in cloth animation [BW98]. While the implicit formulation allows for larger time steps in very stiff equations, they require the solution of a system of non-linear equations per step. The few given experimental timings in the literature indicate that the usefulness of implicit solvers in real-time animation is still limited [BW98, HE01, VMT01], even though the methods compare favorably within the cloth animation domain. Additional difficulties arise when modeling the non-linear stiffness of skin. I have used the explicit Verlet integration scheme since it shows adequate performance and better stability than the Euler method, at similar ease of implementation. On a 1.7 GHz Xeon CPU, the simulation of the mesh depicted in Figure 6.4 achieves stable 50 Hz. Table 6.1 summarizes the spring mesh parameters used in the experiments.

As the statistics in Figure 6.5 show, the simulation (including surface normal updates) takes up less than half of the available time between frames at that rate. Future use of alternative integration methods may be motivated for instance by more complex skin tissue models with a larger stiffness range. An obvious possibility for improving

#skin surface springs	1791	E_s low	100
#skin surface nodes	625	E_s high	1000
#additional springs	625	stiffness skin-muscle springs	10
#additional static nodes	1250	stiffness skin-bone springs	30
node mass	0.31 g	damping coefficent γ	30

Table 6.1: Spring mesh data at a glance, for the reference head.

the integration algorithm is the introduction of variable time steps: larger steps can be taken near stable states. For instance, BARAFF and WITKIN examine the stretch forces of the material at every frame to adjust the step size dynamically [BW98].

The spring mesh structure presented in this chapter incorporates skull non-penetration constraints, and also resists volume compression in the vertical direction. The main benefit compared to local geometric constraints [LTW95] is the absence of an explicit constraint mechanism: no evaluation of skull geometry or skin cell volume has to be performed with each time step. In practice, the mechanism effectively avoids undesirable motion of the skin through the muscle and skull layers, at low computational cost.

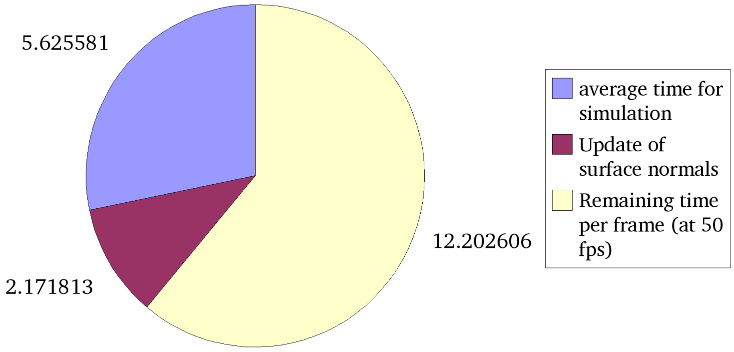


Figure 6.5: Timings for spring mesh update. The values are given in ms, with a total frame time of 20 ms. Also included is the update of surface normals, necessary after skin deformation for proper rendering. Besides rendering, the remaining frame time is spent on muscle updates, cf. Figure 7.21 in Section 7.7.

CHAPTER 7

Facial Muscle Model

The anatomical juxtaposition of two orbicularis oris muscles in a state of contraction.
– Dr. Henry Gibbons, definition of a kiss

Over the past decades, different types of biomechanical models have been developed that reflect the growing body of knowledge about the complex structure of muscles [KS98]. These models focus on the replication of the dynamics of a muscle under isotonic and isometric contractions, i.e., the changes of muscle fiber lengths and generated forces over time (see Section 4.3.1). Isotonic contraction is the most important mode of contraction pertaining to the display of expressions on the face. A virtual muscle model also should support the various muscle shape configurations that exist in the human face, exemplified in Figure 7.1: long and thin strands, broad sheets, curved muscles, and sphincters.

In my muscle model, a muscle is represented as a bundle of fibers, each of which has a piecewise linear polygon as its control structure, similar in spirit to the model developed by LEE *et al.* [LTW95]. Geometric shapes are attached to the segments of this control polygon. Isotonic contraction of a virtual muscle is controlled by a numerical parameter, causing shortening and deformation of the muscle geometry. The simplifying assumption is made that all muscle contractions in the face are carried out instantaneously, i.e., that the internal muscle dynamics and time dependent behavior can be neglected.

Fiber-based muscle models are commonplace in the facial animation community. The model I propose here extends the basic concept to incorporate many important features of muscle behavior pertaining to the face, raising the level of realism and expressiveness that can be achieved. I first discuss these features in the following section, before delving into the details of the muscle model implementation in Section 7.2. A solution for the problem of practical design and construction of a set of muscles is demonstrated in Sections 7.4 and 7.5. The chapter concludes with examples and a discussion of the capabilities and limitations of the proposed muscle model.

7.1 Observations on Muscle Contraction

The muscle models used in current physics-based facial animation systems emulate the shortening of single or grouped muscle fibers under isotonic contraction. A closer

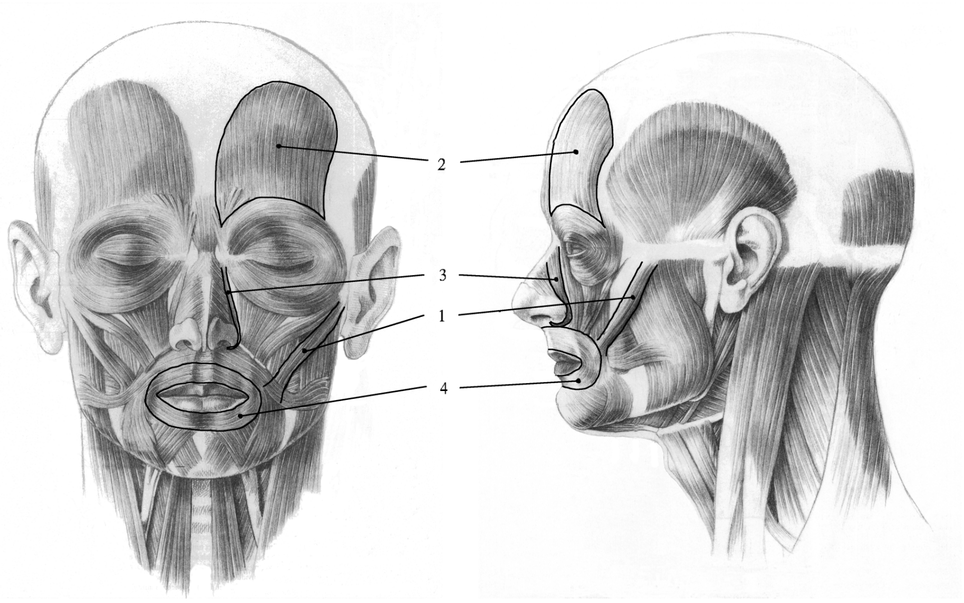


Figure 7.1: Different muscle layouts as they appear in the human face: 1) long and thin strands (zygomaticus major); 2) broad sheets (frontalis); 3) curved muscles (levator labii superioris alaeque nasii) 4) sphincters (orbicularis oris). Image: [SF00]

examination of the behavior of real facial muscles reveals that a number of visible effects resulting from muscle contraction are not supported by simple models:

Interweaving: Muscles are often layered, sliding freely across each other. But, as WATERS and FRISBIE point out [WF95], muscles may also intertwine and merge so that their actions are coupled—a fact that can be observed especially in the region around the mouth. Many of the muscles from the lower face merge into the orbicularis oris, for instance, the zygomaticus and the levator labii superioris.

Elastic deformation: The merging of muscles results in mutual deformation on contraction, i.e., the movement and deformation of a muscle can be caused and constrained by other muscles. It can thus not only actively *contract*, but also be passively *stretched*. This is prevalent in the movements of the mouth, since nearly all of the muscles in the lower face merge into the orbicularis oris. A smile, for instance, caused by contraction of the zygomaticus muscles, pulls at the upper lip part of the orbicularis oris, leading to stretching of this muscle, also elongating and straightening the triangularis muscles (which are fixed at the chin, and also merge into the orbicularis oris).

Bulging: Contracted muscles become thicker; this is very visible in the orbicularis oris when making a pout. In contrast, a muscle becomes thinner when it is stretched.

Independent contraction of fibers: Muscles are composed of many parallel fibers, which mostly contract in unison, shortening by equal amounts. But especially 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 the upper and lower lip are able to move independently. For instance, during the articulation of an “f”, the upper lip protrudes while the lower lip retracts.

These effects suggest a muscle model based on physical properties, coupled with proper numerical simulation, for instance using finite element methods. However, in the context of real-time animation the computational overhead is prohibitive. To produce realistic animation, I thus emulate the listed behaviors by a number of simple geometric deformations, plus a coarse approximation of the network of interconnected muscles using linearly elastic springs.

7.2 Virtual Muscle Representation

The muscles are built from individual fibers that are in turn composed of piecewise linear segments. These fibers can be combined into parallel groups to form sheets of arbitrary width. The surface of the muscle is produced by the union of the geometric shapes that are attached to the segments. Two types of contraction are supported by the model: a *linear muscle* can contract along its length towards one fixed end, while a *sphincter muscle* contracts towards a center point, typically in a direction roughly orthogonal to the direction of the fibers. Both muscle types are essentially described by the same type of control polygon, but a sphincter’s representation includes the specification of the center of contraction. The internal representation of the muscle by control polygon and assigned geometric shape is discussed in this section. Since the process of computing the deformation of a set of interconnected muscles is rather complex, the details of muscle contraction are described separately in Section 7.3.

7.2.1 Muscle Control Polygons

A muscle fiber consists of N control points $\mathbf{p}_i \in \mathbb{R}^3$ $(i = 0, \ldots, N-1)$ forming a control polygon $\mathcal{P}$. A linear muscle always contracts towards $\mathbf{p}_0$. A sphincter muscle has an additional center of contraction $\mathbf{p}^*$, towards which the control points will move. For reproducing protrusion and retraction of the orbicularis oris, a static central point of contraction is not sufficient. Thus, $\mathbf{p}^*$ can move along a central axis $\overline{\mathbf{a}_0\mathbf{a}_1}$, giving the muscle an additional parameter $a \in [0, 1]$ that describes the position of the current contraction center between the axis endpoints $\mathbf{a}_0$ and $\mathbf{a}_1$: $\mathbf{p}^* = (1-a)\mathbf{a}_0 + a\mathbf{a}_1$. Figure 7.2 shows example polygons for a linear and a sphincter muscle. Note that the polygon for a sphincter muscle is not closed, contrary to what one would expect for a circular muscle. This is essential for the modeling of muscles that consist only of a semi-arc. In fact, none of the muscles in the face is really a closed ring: the orbicularis oculi is attached to the nasal root, and the orbicularis oris consists of several parts (see Section 7.6).

7.2.2 Muscle Segment Shapes

Each muscle segment has a local coordinate frame, defining its current orientation, width, length, and height. The axes of this frame are defined by the segment’s direction, as given by connecting the control points, and an additional up vector, pointing

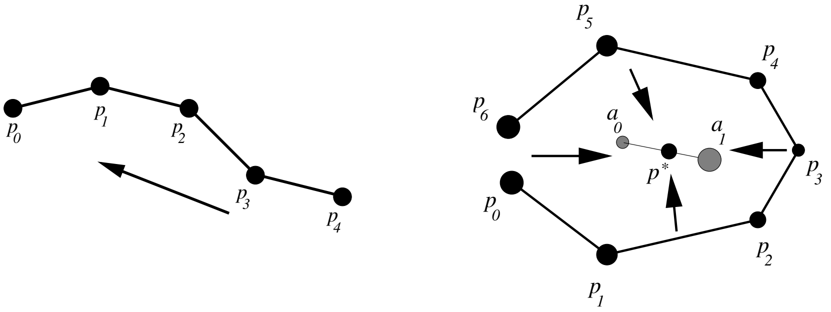
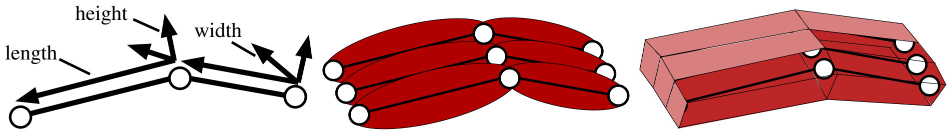


Figure 7.2: Piecewise linear muscle fibers with control polygons $\mathcal{P} = \{p_i\}$. Left: fiber of a linear muscle; right: a sphincter, with contraction center point on the central axis. Arrows indicate general direction of contraction.

towards the skin. The third axis—for the width dimension—is constructed orthogonally to these two. Figure 7.3(a) shows two frames belonging to consecutive segments of a muscle fiber. The frames are used to store the positions of spring mesh nodes that attach to the segments in segment-local coordinates. Deformation of the muscle changes the local frames, and the attached spring mesh nodes are displaced according to their new absolute positions. More details about the connection of skin and muscles are given in Section 7.4.6. The important point here is that the changes of the local coordinate frames during muscle deformation completely define the influence of the muscle onto the skin layer, with no further information about muscle geometry required.

Nonetheless, an actual muscle surface needs to be defined to find the points of attachment for the spring mesh nodes. To this end, a geometric shape is aligned to each segment's local coordinate frame. This segment geometry is used for visualization during interactive construction of a muscle, and for attachment to the spring mesh system. The attachment process as described in Section 7.4.6 requires ray intersection tests, normal computation, and inside / outside tests to be performed on the geometry. Ellipsoids lend themselves easily to these tasks [CW92], but require a certain overlap between neighboring segments to avoid large holes in the muscle surface [KHS01b]. Still, gaps cannot be completely avoided without making the ellipsoids too large, resulting in rare failures of ray intersections or containment tests at the transitions between muscle segments. Figure 7.3(b) shows fibers of a muscle with attached ellipsoids.

These limitations are avoided by using a box-like shape [KHYS02]. For each fiber, a chain of polygonal boxes is constructed, aligned to the frame axes. To obtain a closed surface, the corner vertices of adjacent boxes are unified at the average of their positions. In contrast to the ellipsoid segments, the vertices of the box shapes thus need to be recomputed after a deformation of the muscle if the muscle is to be visualized. Also, ray intersection tests and point inside / outside tests are more complicated than with a quadric surface, requiring testing against the polygonal representation of the box. The advantage is a more well-defined muscle surface without gaps. As stated above, the complexity of the geometric representation of the segments is in any case inconsequential for the run-time performance of the system. Figure 7.3(c) shows fibers of a muscles with attached boxes.



(a) (b) (c)

Figure 7.3: Muscle segment geometry. a) muscle fiber with frames; b) ellipsoids assigned to three neighboring fibers; c) boxes assigned to fibers.

7.2.3 Muscle Connection Constraints

A muscle typically consists of many fibers running in parallel. To ease the management of muscle connections, I use simplified versions of the muscles, essentially representing each as a single fiber. In particular, the following simplifications are made:

- All parallel segments in the control polygons of a muscle's fibers are treated as a union. This union will be referred to as a segment of the *muscle* in contrast to a segment of a *fiber*.
- The positions of parallel fiber control points $\mathbf{p}_i$ are averaged, being denoted as the control point $\bar{\mathbf{p}}_i$ of a *muscle*. For discussion of the muscle connections, it is often necessary to refer to particular muscles. An upper index will be used to denote this, so $\bar{\mathbf{p}}_i^m$ is the i-th control point of muscle m.
- Displacing the control point of a muscle is carried out by applying the same displacement to the individual fiber control points it represents.

The free end of a linear virtual muscle can be connected to a sequence of segments on any other muscle. The connections are automatically detected by testing the shapes assigned to the end segment of each muscle for intersection with the shapes of all other muscle segments during initialization. Only the end segments are considered for testing to allow muscles to freely cross each other without interaction. For each of the S connected components in the segment intersection graph, a *constraint group* with index $k, 0 \leq k < S$, is set up. Each segment is bounded by the two muscle control points at either end, and for each pair of segments in group k the two points closest to each other are identified. The set $\mathcal{C}_k \subset \mathbb{N}^2$ references these pairs of nearby control points by their muscle and point index: $(m, i) \in \mathcal{C}_k$ refers to the position $\bar{\mathbf{p}}_i^m$. The initial distance of each point in group k to the group's centroid is computed as well, denoted by $d_k^{m,i}$. An example showing three muscles merging into a fourth muscle is shown in Figure 7.4.

Once the constraint groups have been formed, a simple mass-spring system, approximating the elastic behavior of the network of intertwined muscles, is constructed. An example for a spring mesh constructed in this way is shown in Figure 7.5. This system is set up in four steps:

1. Mass nodes are created for all $\bar{\mathbf{p}}_i^m, (m, i) \in \mathcal{C}_k$, contained in each constraint group k (cf. the green and red dots in Figure 7.5).
2. The nodes that belong to the same muscle are connected by a spring, thereby linking constraint groups to each other (cf. the red lines in Figure 7.5).

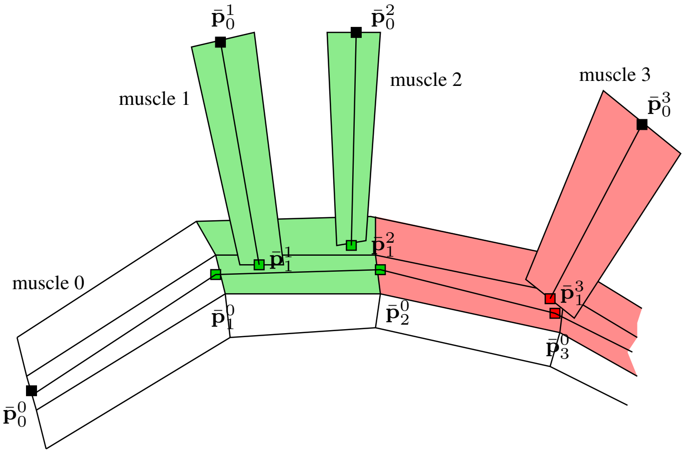


Figure 7.4: Formation of constraint groups. The muscles 1,2, and 3 merge into muscle 0. Two constraint groups are formed by intersecting segment shapes: group 0 (green) and group 1 (red); the *muscle* control points $\bar{\mathbf{p}}_i^m$ are averaged from the *fiber* control points. The control point indices are collected for each group: $\mathcal{C}_0 = \{(0,1),(0,2),(1,1),(2,1)\}$ and $\mathcal{C}_1 = \{(0,3),(3,1)\}$.

3. For every linear muscle whose end control point is part of a constraint group a static node is created at the position of the first muscle control point (the fixed bone attachment). A spring is inserted connecting this fixed node to the end point (cf. the black dots and lines in Figure 7.5).

4. Within each group, the nodes corresponding to points belonging to different muscles are connected by springs (cf. the green lines in Figure 7.5).

7.3 Muscle Deformation

For each frame of an animation, the current contraction of each muscle is specified as a scalar parameter $c \in [0,1]$, where $c = 0$ means no contraction and $c = 1$ full contraction. The sphincter parameter $a \in [0,1]$ may be specified to move the center of contraction for that muscle. The muscles always drive the skin deformation and are not in turn moved by the skin. However, some of the muscles are attached to the mandible, following its rotation. The jaw rotation angle is thus set before updating the muscles. In reality, those muscles are pulled downwards by the skin tissue they are embedded in, which is in turn attached to the jaw bone.

The process of translating muscle contraction parameters into deformation of the muscle and finally displacing the attached skin nodes is carried out in separate stages. The first three steps work entirely on the control polygons, then the local frame of each segment is updated, and finally the influence on the skin tissue layer is computed:

1. Deform each muscle fiber according to current contraction values and jaw rotation.

2. Satisfy muscle connection constraints.

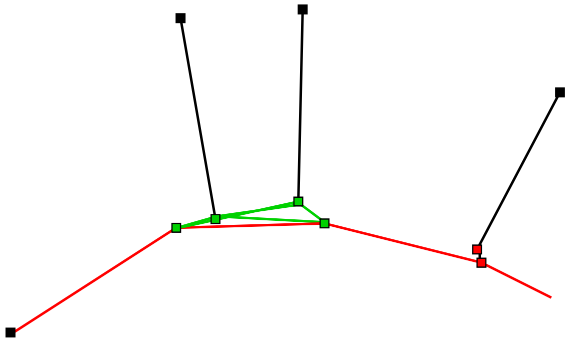


Figure 7.5: The spring mesh constructed to connect the two constraint groups from Figure 7.4. Black nodes are static points, red and green ones correspond to the muscle control points in the constraint groups. Line colors red, black, and green correspond to steps 2,3, and 4 as explained in Section 7.2.3.

3. Relax unconstrained parts of the muscle that have been stretched.
4. Compute shape changes due to bulging and stretching.
5. Update attached spring mesh nodes.

The following Sections 7.3.1 to 7.3.5 explain these steps in detail.

7.3.1 Evaluating Jaw Rotation and Muscle Contraction

Each muscle is attached either to the skull or to the jaw. This is done automatically during muscle construction, as discussed in Section 7.4.5. If a muscle is attached to the jaw, its initial control points $\mathcal{P}$ will be rotated around the rotational axis of the jaw by the current angle. To avoid unnecessarily complicating the notation, it is silently assumed in the following that the lower face control polygons have been transformed in this way, before the contraction of the fibers is computed.

For a linear muscle fiber, given the contraction value c, a new control polygon $\mathcal{Q} = \{\mathbf{q}_i\}_{i=0}^{N-1}$ is computed from $\mathcal{P} = \{\mathbf{p}_i\}_{i=0}^{N-1}$. To this end, each control point $\mathbf{p}_i \in \mathcal{P}$ is assigned a parameter $t_i \in [0,1]$ describing the point's position on the polygon:

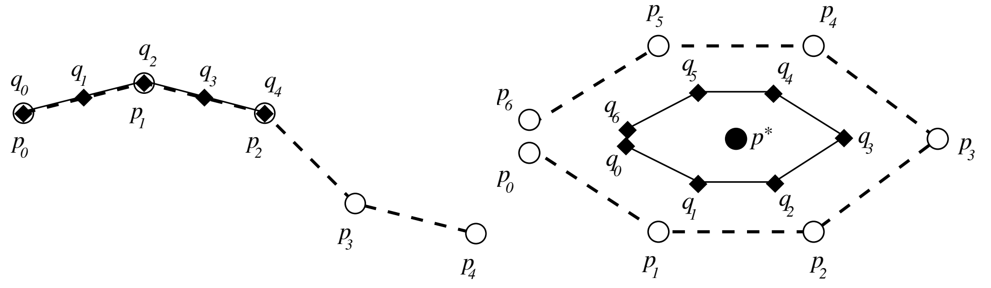
$$t_i := \begin{cases} 0 & \text{, if } i = 0, \\ \frac{\sum_{j=1}^{i} \|\mathbf{p}_j - \mathbf{p}_{j-1}\|}{\sum_{j=1}^{N-1} \|\mathbf{p}_j - \mathbf{p}_{j-1}\|} & \text{, else.} \end{cases}$$

The parameters t_i are scaled by the contraction factor $(1-c)$ and then clamped to $[0.01, 1]$ to avoid shrinking a segment too much:

$$\tilde{t}_i \ := \ \max\{(1-c)t_i, 0.01\}.$$

Next, each parameter $\tilde{t}_i$ is mapped to the index $k_i \in \{0, \ldots, N-2\}$ of the starting point of the segment that contains $\tilde{t}_i$:

$$k_i := \begin{cases} 0 & \text{, if } i = 0, \\ j \ : \ t_j < \tilde{t}_i \leq t_{j+1} & \text{, else.} \end{cases}$$



(a) (b)

Figure 7.6: Contraction ($c = \frac{1}{2}$) of a linear (a) and a sphincter (b) muscle fiber. The control points $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$ represent the relaxed and contracted muscle, respectively.

Finally, the new control points $\mathbf{q}_i$ are computed by linear interpolation:

$$\mathbf{q}_i \ := \ \mathbf{p}_{k_i} + (\mathbf{p}_{k_i+1} - \mathbf{p}_{k_i}) \frac{\tilde{t}_i - t_{k_i}}{t_{k_i+1} - t_{k_i}}.$$

The net result is a shortening of the fiber by a factor of c, where the control points move along the path described by the original control polygon, as shown in Figure 7.6(a).

For sphincter muscles, segments are simply contracted towards the current center point $\mathbf{p}^* \in \mathbb{R}^3$ (see Figure 7.6(b)):

$$\mathbf{q}_i \ := \ \mathbf{p}^* + (1 - c)(\mathbf{p}_i - \mathbf{p}^*).$$

To accommodate for the non-homogeneous contraction of the orbicularis oris, the fibers of a sphincter are allowed to protrude by different amounts. The outer part of the orbicularis oris (directly beneath the nose) protrudes very little, since this part is constrained by other muscles and the skull attachment of the surrounding tissue. On the other hand, the inner lip part can move freely. This is realized by varying the center of contraction on each fiber. Points on the outmost fiber contract towards the point $\mathbf{p}^\perp$ found by direct projection of $\mathbf{p}_i$ onto the vector $\mathbf{a} = \mathbf{a}_1 - \mathbf{a}_0$:

$$\mathbf{p}^\perp = \mathbf{a}_0 + \left(\frac{\mathbf{a} \cdot (\mathbf{p}_i - \mathbf{a}_0)}{\|\mathbf{a}\|^2} \right) \mathbf{a}$$

The innermost fiber points use $\mathbf{p}^*$. For a muscle with $f > 1$ fibers running in parallel, the fibers are numbered from 0 to $f - 1$, starting at the outmost fiber. The contraction center $\mathbf{p}_j^*$ used by the points on fiber j is found by linear interpolation between the extreme points:

$$\mathbf{p}_j^* = \frac{j}{f-1}\mathbf{p}^* + \left(1 - \frac{j}{f-1}\right)\mathbf{p}^\perp.$$

The effect is shown in Figure 7.7.

7.3.2 Keeping Muscles Connected

After deformation due to muscle contraction has taken place, the muscle control points in the constraint groups are not located closely together anymore. This is corrected in a two-step procedure.

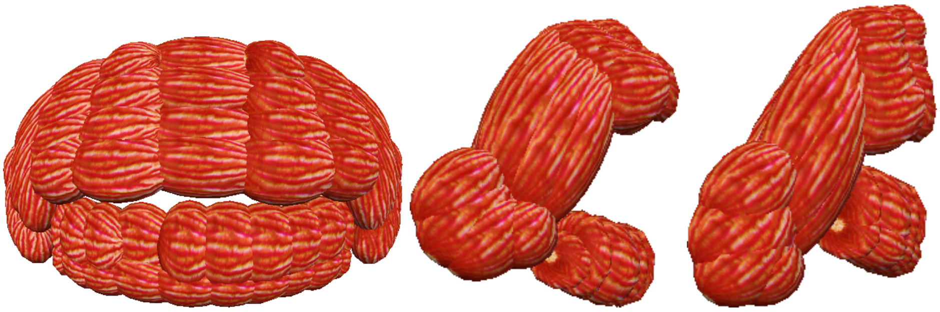


Figure 7.7: Two-part orbicularis oris model. Left: relaxed state with closed mouth, front view. Center: protruded lips, slightly opened mouth, side view. Right: upper lip retracted, lower lip moved upward and inward.

Geometric Constraint Resolution

First, the point positions in each constraint group k are displaced to regain their initial distance from the group center. The current group center $\mathbf{g}_k$ is computed as a weighted average of the control points $\bar{\mathbf{q}}_i^m$, $(m, i) \in \mathcal{C}_k$, where muscles with a higher contraction value also have a higher weight, thereby drawing the new center point towards them:

$$\begin{aligned} w_k &= \sum_{m \in \mathcal{M}_k} 1 + \lambda c_m, \\ \mathbf{g}_k &= \sum_{(m,i) \in \mathcal{C}_k} \left(\frac{1 + \lambda c_m}{w_k} \bar{\mathbf{q}}_i^m \right), \end{aligned}$$

where $\mathcal{M}_k$ is the set of different muscle indices in $\mathcal{C}_k$, and c_m denotes the contraction value of muscle m. The formulation guarantees that for a set of completely relaxed muscles ($c_m = 0$), the group center $\mathbf{g}_k$ will be located in the centroid of the control points. The influence of the muscle contraction is scaled by the parameter λ, which can be viewed as a measure for the increasing stiffness of the muscles under contraction: a value of $\lambda = 0$ evenly distributes the influence of each muscle, regardless of its contraction state. Higher values let the more strongly contracted muscles exert a higher "force" onto $\mathbf{g}_k$. A value of $\lambda = 30$ is used in practice.

Given the new center point, the $\bar{\mathbf{q}}_i^m$ are moved directly towards it to restore the initial distance:

$$\forall (m, i) \in \mathcal{C}_k : \bar{\mathbf{q}}_i^m \leftarrow d_k^{m,i} \frac{(\mathbf{g}_k - \bar{\mathbf{q}}_i^m)}{\|\mathbf{g}_k - \bar{\mathbf{q}}_i^m\|}$$

This mechanism implements the desired behavior that highly contracted muscles exert a stronger pull on the other muscles connected to it. Figure 7.8 demonstrates this for one constraint group.

Dynamic Constraint Resolution

For smaller muscle contractions, the above geometric method suffices to keep the muscles connected. With stronger deformations, it does not warrant good connection of the merged muscles, though. Also, the effect of one muscle pulling at another muscle is too local: the deformation at one constraint group should also have an influence on

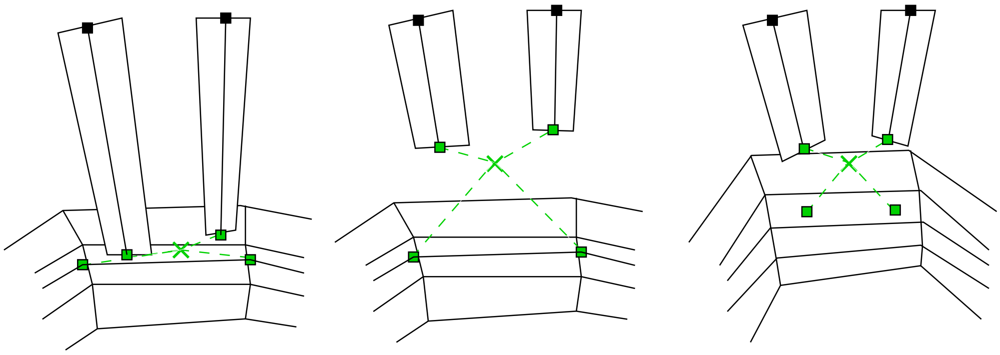


Figure 7.8: Geometric constraint resolution for the constraint group 0 from Figure 7.4. Left: Muscle control points and center (green cross) at rest, with distance from points to center (green lines). Center: The two top muscles are contracted, displacing the center of the group. Right: Muscle control points have been moved to restore original distances.

distant parts of the muscle. To get an approximation of this type of elastic muscle behavior, the muscle configuration resulting from the geometric constraint resolution step serves as an initial guess for a global constraint resolution based on dynamic simulation of the spring mesh described in Section 7.2.3.

The node positions of the spring mesh are updated according to the new muscle control points resulting from the geometric constraint resolution. The rest length of the springs is set to the current distance between their end nodes, representing the preferred length of each muscle. A numerical simulation, similar to the one described in Chapter 6, distributes forces across the linked muscle segments. Each run of this simulation spans the time between two rendered animation frames. All mass nodes start from their rest state each time, so there is no continuous update over the course of the animation. The spring mesh is typically very small—in the implemented muscle set it consists of $N_n = 70$ nodes and $N_s = 143$ springs. The numerical simulation is thus carried out quickly and takes up only a small fraction of the overall computation time. See the statistics in Section 7.6 for details. The spring mesh parameters are defined with a node mass of $m_i = 0.01$ g $(i = 0, \dots, N_n - 1)$, a damping factor $\gamma = 3$, and spring stiffness $k_s = 40$ $(s = 0, \dots, N_s - 1)$.

7.3.3 Muscle Relaxation and Stretching

When control points on a muscle are pulled at by other muscles, the lengths of the fiber control polygons change. To simulate stretching of an elastic muscle, the control polygons are straightened out according to the elongation. The control points that are part of constraint groups are held fixed in this step, but each section between two groups along the length of the muscle is modified, including the sections towards the muscle ends.

Let u and v be indices pointing to neighboring constrained control points on a fiber. If this muscle section is fully stretched, the current points $\mathbf{q}_i, u < i < v$ are projected onto the line connecting $\mathbf{q}_u$ and $\mathbf{q}_v$, as illustrated in Figure 7.9. The target positions $\mathbf{t}_i$ on this line are computed by assigning parameters in the interval $[0, 1]$ to the control

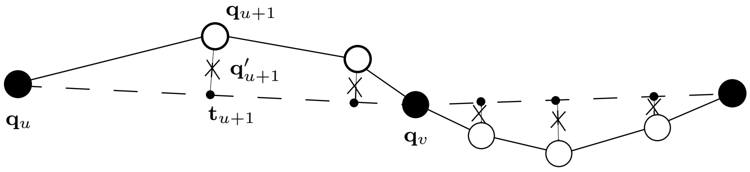


Figure 7.9: Muscle relaxation of two neighboring constrained muscle fiber sections. The black nodes are part of a constraint group, the white nodes are unconstrained. The muscle is straightened in the unconstrained parts by projecting control points onto a line.

points and mapping them onto the line segment:

$$\mathbf{t}_i := \begin{cases} \mathbf{q}_u & \text{, if } i = u, \\ \mathbf{q}_u + (\mathbf{q}_v - \mathbf{q}_u)\frac{\sum_{j=u+1}^{i} \|\mathbf{q}_j - \mathbf{q}_{j-1}\|}{\sum_{j=u+1}^{v} \|\mathbf{q}_j - \mathbf{q}_{j-1}\|} & \text{, else.} \end{cases}$$

The new position for control point i is linearly interpolated between $\mathbf{q}_i$ and $\mathbf{t}_i$, according to the elongation of the fiber. First the rest length l_r and the current length l_c of the “closed muscle loop” is computed, i.e., the length of the control polygon if its end points were connected:

$$\begin{aligned} l_r &= \sum_{i=1}^{N-1} \|\mathbf{p}_i - \mathbf{p}_{i-1}\| + \|\mathbf{p}_{N-1} - \mathbf{p}_0\| \\ l_c &= \sum_{i=1}^{N-1} \|\mathbf{q}_i - \mathbf{q}_{i-1}\| + \|\mathbf{q}_{N-1} - \mathbf{q}_0\| \end{aligned}$$

These lengths are used to compute a weight w for the linear interpolation, which is clamped to the range $[0, 1]$:

$$w = \max\left(1, \min\left(0, \frac{l_c - l_r}{l_r}\mu\right)\right)$$

If the muscle is relaxed and not stretched at all, w equals zero. On the other hand, as the loop l_c becomes bigger due to pulling at the control vertices of the muscles, the value approaches 1.0, which is reached when $l_c = 2l_r$. The parameter μ is used to model the stiffness of the muscle: the higher the value, the more does the muscle fiber respond to the stretching stress. In practice, I use $\mu = 4$. Now the interpolation is performed to displace the control points:

$$\mathbf{q}'_i \leftarrow \mathbf{q}_i(1 - w) + \mathbf{t}_i w.$$

In this formulation, muscle contraction counteracts stretching, since l_c becomes smaller for a shortened control polygon. A contracting muscle will thus still closely follow the path prescribed by its control polygon, while muscles that stretch are straightened out.

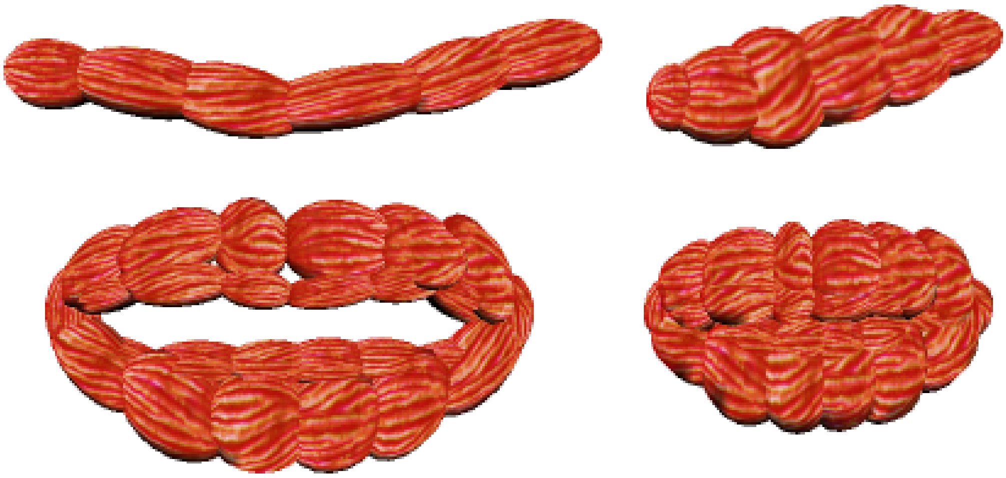


Figure 7.10: Relaxed (left) and contracted (right) muscles: a single fiber (top) and a sphincter (bottom) modeling the *orbicularis oris*

7.3.4 Updating Muscle Geometry: Bulging and Thinning

With the previous steps, the muscle fiber control polygons have assumed their final form for the current animation frame. The last step in deformation of the muscles modifies the segment geometry to simulate bulging and thinning. Real muscles get thicker on contraction and thinner on elongation; simulating this behavior enhances visual realism: when the face smiles, the lips retract slightly as they stretch. On the other hand, the lips get a little thicker, when the mouth forms an “o” or a kiss. For a linear muscle, its center should exhibit the highest bulge, corresponding to the belly of a real muscle. Sphincters on the other hand bulge evenly in my model.

Bulging is achieved by scaling the height of each muscle segment $\overline{\mathbf{q}_i\,\mathbf{q}_{i+1}}$ by a bulge factor b computed from the change in muscle length:

$$b := 1 + \nu \left(1 - \frac{\sum_{i=1}^{N-1} \|\mathbf{q}_i - \mathbf{q}_{i-1}\|}{\sum_{i=1}^{N-1} \|\mathbf{p}_i - \mathbf{p}_{i-1}\|} \right)$$

Choosing $\nu = 5$ as a reasonable value, a muscle that shortens to 75% of its original length is scaled in height by $b = 2.25$. In contrast, if the muscle is stretched to 125%, the scale factor is $b = -0.25$, thinning the muscle by $1/4$ of its height. For a sphincter muscle, b is directly used in this manner to scale the up axis of all segments.

For each linear muscle with at least three segments, the bulge is modulated per segment with a simple quadratic function that vanishes over the first and last segment and has a maximum value of 1.0 over the central segment. In this case, the per-segment bulge factor b_i is computed as

$$b_i := 1 + \nu \left(1 - \frac{\sum_{i=1}^{N-1} \|\mathbf{q}_i - \mathbf{q}_{i-1}\|}{\sum_{i=1}^{N-1} \|\mathbf{p}_i - \mathbf{p}_{i-1}\|} \right) \left[1 - \left(\frac{2i}{N-2} - 1 \right)^2 \right].$$

Figure 7.10 illustrates the effect of this simple heuristic for sphincter and linear muscles.

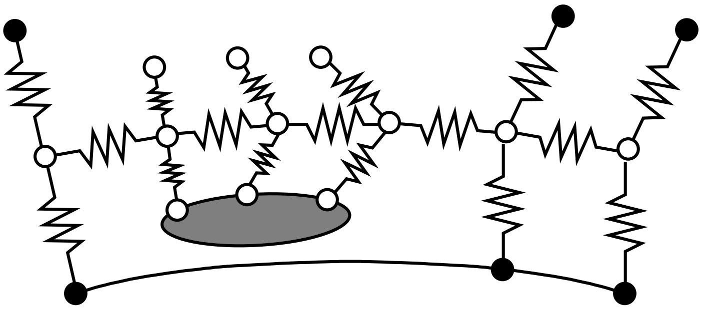


Figure 7.11: Update of attached spring nodes on muscle contraction. Mass points moving due to the contraction are marked by ∘. See also Figure 6.4 in Chapter 6.

7.3.5 Updating Attached Spring Nodes

The movement and deformation of the fiber segments is used to build a new local frame for each segment. To this end, the frame associated with the original line segment $\overline{\mathbf{p}_i\,\mathbf{p}_{i+1}}$ is transformed to the new segment $\overline{\mathbf{q}_i\,\mathbf{q}_{i+1}}$ using an affine transformation, incorporating translation, rotation, and scaling. The translation vector $\mathbf{t}_i$ is simply the offset between the old and the new center of the segment:

$$\mathbf{t}_i := \frac{1}{2}(\mathbf{q}_i + \mathbf{q}_{i+1}) - \frac{1}{2}(\mathbf{p}_i + \mathbf{p}_{i+1}).$$

The rotational part is the smallest rotation that aligns $\mathbf{v}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$ to $\mathbf{w}_i = \mathbf{q}_{i+1} - \mathbf{q}_i$:

$$\begin{aligned} \mathbf{a}_i &:= \mathbf{v}_i \times \mathbf{w}_i, \\ \theta &:= \operatorname{acos}\left(\frac{\mathbf{v}_i \cdot \mathbf{w}_i}{\|\mathbf{v}_i\| \cdot \|\mathbf{w}_i\|}\right), \end{aligned}$$

where $\mathbf{a}_i$ is the axis of rotation, and θ its angle. Finally, the new frame is scaled to the length of $\overline{\mathbf{q}_i\,\mathbf{q}_{i+1}}$ and the height that resulted from the bulging step. This new segment transformation is now applied to the attachment points stored in frame-local coordinates with each segment. The resulting world space displacement of each node is applied to its mirror node (see Figure 7.11 and Section 6.4). As a simple overrelaxation scheme, the displacement is also applied to the node on the skin surface, situated between the spring attaching to the muscle surface and its mirror. This movement is usually too violent, since the skin does typically not follow the muscle contraction completely. But as a first estimate, it helps to speed up the convergence of the simulation, which is run for one time step after the spring mesh nodes have been updated as described here (see Chapter 6).

Only for visualization of the muscle, the fiber segment transformation is also applied to the associated segment shape. Normally, the shape itself does not need to be updated during animation.

7.4 Building Muscles from Geometry

The previous sections discussed how virtual muscles are represented and deformed during animation. It is in principle possible to manually put together muscles by laying down control polygons and specifying segment shapes. For any practical system,

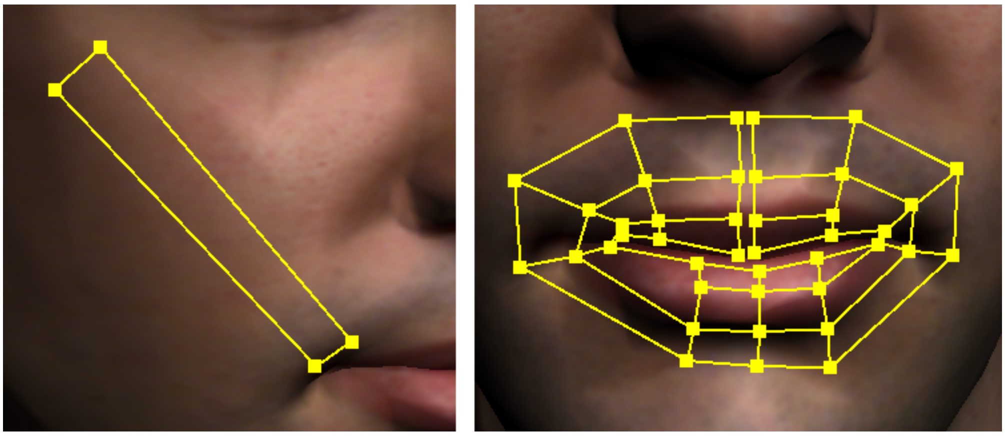


Figure 7.12: A simple grid (left, zygomaticus major) and a non-uniform complex grid (right, orbicularis oris).

though, the construction of muscles must be aided by the computer. In this section, I describe a method that allows automatic construction of a set of muscles based on given facial geometry, for instance, a human head model created from range scan data. Since the muscles of the face lie closely underneath the skin and have a great influence on the shape of the surface, it is possible to derive the shape of the muscles from the surface geometry. This approach is used to interactively sketch muscles as coarse outlines onto a polygonal head model, described in Section 7.5. This interactive method has to be used only for the initial design of the muscle structure on the reference head model (see Section 5.1). When the reference model is adapted as a whole to given target geometry, the muscle outlines are automatically adapted without further user intervention. The actual muscle shape is always constructed from these outlines by the algorithm explained in the following.

7.4.1 Muscle Shaping Overview

The outline is defined on the facial geometry and can be of arbitrary detail, from the minimum of a quadrilateral up to highly complex grids, as shown in Figure 7.12. Given the skin mesh and such a muscle grid, an optimization procedure determines the following parameters needed to create the muscles:

- the number of muscle fibers;
- the number of segments per fiber;
- width, height, and length of each segment;
- position of each fiber control point;
- alignment of each segment shape coordinate system.

The skin thickness τ^s, assumed to be constant over the whole face, and the minimum and maximum muscle layer thickness τ^m_{min} and τ^m_{max} are given as additional input parameters to the optimization step. These parameters depend on the individual anatomy of the model and have to be adjusted by the user. For the reference head model, for instance, the values $\tau^s = 2$ mm, $\tau^m_{\text{min}} = 2$ mm, $\tau^m_{\text{max}} = 6$ mm have been chosen.

In the optimization step, the muscle is built to fit underneath the skin surface, within a prescribed distance range. The process is also guided by the resolution of the skin mesh: too highly refined muscles just add to the computational load during animation without enhancing animation quality, while muscles that are too coarse may not be following the surface closely enough to result in realistic deformations. A muscle is created from its grid by a four-step procedure:

1. **Grid initialization:** The initial outline is converted into a regular grid, i.e., all rows are assigned the same number of grid points. The grid points are then projected onto the face mesh and placed slightly underneath the skin surface.

2. **Grid refinement:** The grid is adaptively refined until a satisfactory approximation is found.

3. **Muscle creation:** Muscle fibers are created and aligned to the refined grid.

4. **Muscle attachment:** The muscle is attached to the spring mesh, and the control points of the muscle segments are attached to either the skull or the jaw.

These steps are detailed in the following Sections 7.4.2 to 7.4.5.

7.4.2 Grid Initialization

To obtain a regular grid, first the maximum number N_{max} of grid points per row is determined. Then, additional grid points are inserted by linear interpolation into every row that contains less than N_{max} points. This is followed by estimation of normals at the grid points. For the variety of possible grid layouts, I obtained best results by first computing the normal of the balancing plane through the four corner points of each grid cell and then averaging the normals of all adjacent cells at each grid point.

Having computed the grid point normals, the triangles of the face mesh that intersect the projection of the grid onto the skin surface are found and cached for fast lookup during the iterative refinement procedure. The initial grid points are now displaced along their normal direction to lie below the skin in an initial distance of $\tau^s + (\tau^m_{\text{min}} + \tau^m_{\text{max}})/4$, i.e., at the center of a muscle of average thickness running through the cell.

7.4.3 Grid Refinement

The fitting algorithm proceeds by sampling the distances from each cell to the skin surface. Each cell is examined and subdivided if necessary. The grid points are then again displaced to lie within the prescribed distance range below the surface. Simultaneously, the cell thickness is adjusted within the bounds τ^m_{min} and τ^m_{max}. This process is repeated until no more subdivisions are necessary or the cells become to small for further refinement. The main loop of this iteration is organized as follows:

```
repeat
  for each grid cell $c$
    ($d_{\mathrm{min}}$, $d_{\mathrm{max}}$, $\mathbf{p}_{\mathrm{near}}$, $\mathbf{p}_{\mathrm{far}}$) =
        minMaxDistancesToMesh($c$);
    ($e_{\mathrm{near}}$, $e_{\mathrm{far}}$) =
        minMaxError($d_{\mathrm{min}}$, $d_{\mathrm{max}}$, $\tau^s$, $\tau^m_{\mathrm{min}}$, $\tau^m_{\mathrm{max}}$);
    if ($e_{\mathrm{near}}$ == 0 and $e_{\mathrm{far}}$ == 0)
      $c$.thickness = 2($d_{\mathrm{min}}$- $\tau^s$)
    else if ($e_{\mathrm{far}} > e_{\mathrm{near}}$)
      trySubdivisionAtPoint($c$, $\mathbf{p}_{\mathrm{far}}$);
    else
      trySubdivisionAtPoint($c$, $\mathbf{p}_{\mathrm{near}}$);
  moveNewGridPoints();
until no more changes to grid.
```

The procedure `minMaxDistancesToMesh()` returns the two points $\mathbf{p}_{\mathrm{near}}$,$\mathbf{p}_{\mathrm{far}}$ that are projections of mesh points nearest to and farthest away from the cell. To find these points, the grid cell is adaptively subsampled, and a ray from each sample position is shot in the direction of the associated grid normal vector, which is bilinearly interpolated from the cell corners. The base points of the rays with the nearest and farthest intersection points with the cached surface area are returned as $\mathbf{p}_{\mathrm{near}}$ and $\mathbf{p}_{\mathrm{far}}$ along with their signed distance values d_{min} and d_{max}. Both points can be positioned below (positive distance value) or above the skin surface (negative value), see Figure 7.13. The sampling density over the grid cell adjusts to the size of the cell and the number of cached triangles to ensure a minimum number of samples per triangle.

In procedure `minMaxError()`, two distance error values e_{near} and e_{far} are com-

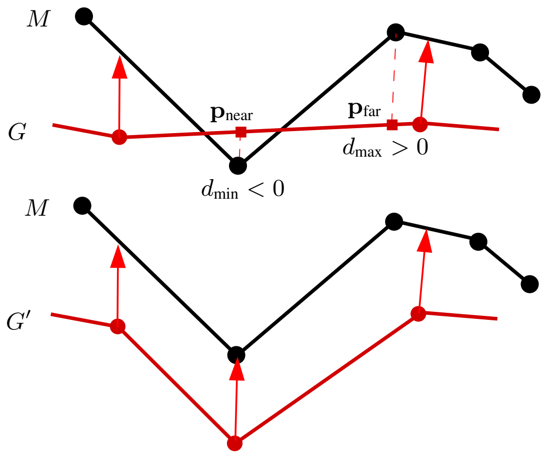


Figure 7.13: Refinement step for a single grid cell (simplified two-dimensional view). Top: Points of grid G have been placed below the skin mesh M along their associated normals. The closest point of the grid cell lies above, the farthest point lies below the skin mesh. Bottom: G has been subdivided at the point of larger error $\mathbf{p}_{\mathrm{near}}$ (see also Figure 7.14).

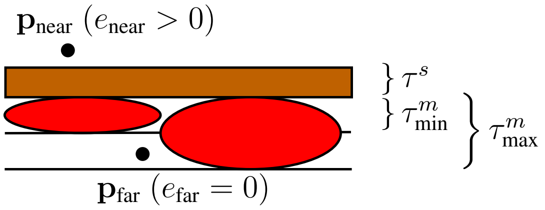


Figure 7.14: The range of thickness for muscle shapes. Below the skin layer of constant thickness τ^s muscles can be inserted with a thickness in the range $[\tau^m_{\text{min}}, \tau^m_{\text{max}}]$. Error values for two exemplary points are shown: $\mathbf{p}_{\text{near}}$ is outside the allowed range for muscle segments and should be moved downwards ($e_{\text{near}} > 0$). $\mathbf{p}_{\text{far}}$ is within range and need not be moved ($e_{\text{far}} = 0$).

puted for the points $\mathbf{p}_{\text{near}}$ and $\mathbf{p}_{\text{far}}$, respectively. The values represent the unsigned distances that $\mathbf{p}_{\text{near}}$ and $\mathbf{p}_{\text{far}}$ would have to be pushed along their grid normal to move them into the allowed range of distances $[\tau^s + \tau^m_{\text{min}}, \tau^s + \tau^m_{\text{max}}]$ below the skin surface (see Figure 7.14). A zero value means that the point is already in this range. Subdivision of a cell is performed when e_{near} or e_{far} have non-zero values, i.e., a point of the cell is too close to or too far away from the skin. Also, if $|d_{\text{max}} - d_{\text{min}}| > \tau^m_{\text{min}}$, the distance from the cell to the skin varies so widely over the cell area that there is enough vertical space for insertion of a thin muscle patch. In this case, $(e_{\text{near}}, e_{\text{far}})$ are set to $(|d_{\text{min}}|, |d_{\text{max}}|)$, causing a subdivision of the cell in the next step.

The procedure `trySubdivisionAtPoint()` is called with the sample position corresponding to the point with the larger error. A new row and/or column through that position is inserted into the grid. Before subdividing a cell along one of its dimensions, I compare the sizes of the resulting sub-cells with the average extent of the cached triangles in that direction. If the sub-cell on one side of the division line would become too small, the insertion point is adjusted to make it bigger, thus reducing the size of the other sub-cell. If the cell is already too small to allow for adjustment, no further subdivision along this direction is performed. These criteria effectively ensure that no very small cells are created that are not sampled densely enough for making distance measurements.

Finally, in `moveNewGridPoints()` the grid points inserted by subdivision are projected onto the surface mesh and positioned at a distance of $\tau^s + (\tau^m_{\text{min}} + \tau^m_{\text{max}})/2$ underneath the skin.

7.4.4 Muscle Creation

After a grid has been refined sufficiently, a sheet of muscle fibers is built. One muscle fiber is inserted longitudinally into each stripe of grid cells, creating one muscle segment per cell. The size of each segment shape is scaled to fill the surrounding cell. When ellipsoids are used, their width and length is slightly enlarged to provide some overlap across cell boundaries. Figure 7.15 shows the creation of a sheet muscle from a simple grid with ellipsoid segment geometries.

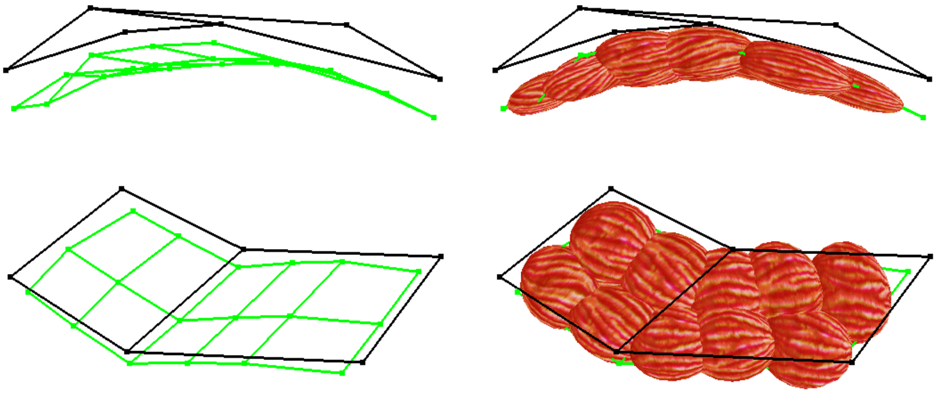


Figure 7.15: Side and top view of a coarse (black) and refined (green) grid and the muscle created from it.

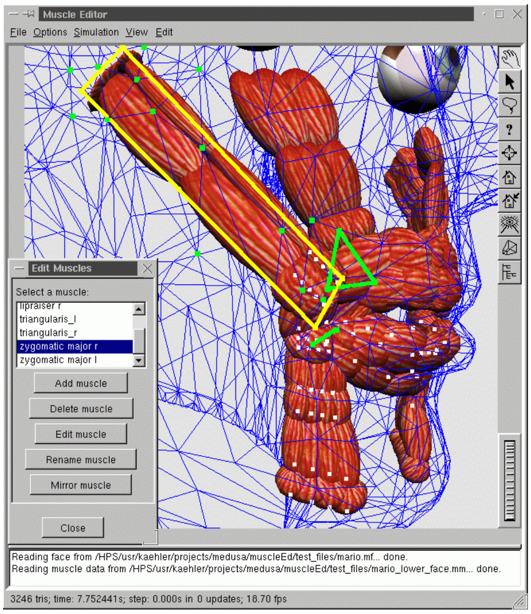
7.4.5 Attaching Muscles to the Skull

To find out if a muscle control point should move along with the jaw (see Section 7.3), rays are sent from the grid points outwards, along their normals, through the skin mesh. The attachment of the closest skin vertex in the hit triangle is examined. If the majority of points in a grid row is closest to jaw-attached skin vertices, the corresponding muscle attachments will also be attached to the jaw. Not all regions of the face have bones underneath, for instance, the lips and the cheeks. Skin vertices in these regions are thus not attached to the bone structure. To decide about the muscle attachment in these cases, the topological neighborhood of the skin mesh is iteratively grown around the intersection point until a bone-attached skin node is found. Using this technique, the upper part of the orbicularis oris is properly assigned to the immovable skull, while the lower part is attached to the jaw.

7.4.6 Attaching Muscles to the Skin

Muscles have to be connected to the spring mesh, so that contraction will influence nearby skin vertices. The vertices within a specified radius of influence from the muscle fibers are considered as candidates for muscle attachment: for each of these skin nodes, the closest point on the surface of the muscle sheet is computed, and a spring is inserted, connecting the skin node with that point. An additional spring is created by mirroring the attachment point at the skin node.

There are special cases where the distance-based computation of attachment points is not sufficient. For instance, when the face mesh has a closed mouth, vertices along the cut separating the upper and lower lip will have almost—if not exactly—the same coordinates. These vertices may thus be attached to the muscles around the upper and lower lips in a nondeterministic way. This will likely cause the upper lip to move along with the lower orbicularis oris and vice versa. To solve this problem, the distance value of each skin node is weighted with the dot product $\mathbf{n}_s \cdot \mathbf{n}_d$, where $\mathbf{n}_s$ is the surface normal at the skin vertex, and $\mathbf{n}_d$ is the normalized vector pointing from the potential attachment point to that vertex. Thereby muscle segments that lie directly below a skin vertex are favored.



```
Reading face from /HPS/usr/kaehler/projects/medusa/muscleEd/test_files/mario.mf... done.
Reading muscle data from /HPS/usr/kaehler/projects/medusa/muscleEd/test_files/mario_lower_face.mm... done.
```

```
3246 tris; time: 7.752441s; step: 0.000s in 0 updates; 18.70 fps
```

Figure 7.16: Visual information while interactively editing a muscle. The muscle grid of the currently edited muscle (yellow); the skin vertices influenced by this muscle (green dots); muscle control points attached to the jaw (white dots); merged muscle segments (connected by green lines).

7.5 Interactive Muscle Editing

Before a muscle can be built from an outline on the face mesh, this outline has to be created first. For editing muscle grids and controlling the resulting muscle layout, I have developed an interactive muscle editor, depicted in Figure 7.16. The essential features of the muscle editor are:

Load face mesh: The first step is loading a triangle mesh into the editor.

Add linear muscle: A muscle is laid out by specifying at least four grid points, starting from the fixed end (attached to bone), and continuing row by row. The muscle grid is refined automatically to fit the geometry and the muscle is inserted, also connecting it to other existing muscles it intersects. The muscle is then fully functional for immediate testing and re-editing.

Add sphincter muscle: Sphincter muscles are created in much the same fashion. There is no fixed end here, so the control point specification can start at either end of the muscle. In addition to the control points, the central axis for contraction needs to be specified. The initial center point of the axis is found by simply clicking on a point on the mesh near the desired contraction center. The normal to the average plane through the muscle control points is used as the axis direction. The generated axis endpoints can be re-edited for precise placement.

Delete muscle: A muscle can be removed completely. This requires recomputation of the spring mesh attachments and the inter-muscle connections.

Edit control points: The control points of a muscle and the end points of the central axis of a sphincter can be moved around arbitrarily for corrections.

Name muscle: For identification by the system, each muscle needs to have a unique name.

Flip muscle: This reverses muscle direction, in case fixed and free ends have been confused by the user during muscle creation.

Mirror muscle: The muscle grid is mirrored about the $x = 0$ plane, and a new muscle is created. This is a convenient shortcut for muscle creation, exploiting the symmetry of most head models. This way, only one half of the muscle set needs to be specified.

Refine grid: The control grid can be refined along its rows and / or columns to create more control points. This gives more freedom over the grid layout without the need to recreate the whole muscle from scratch.

Save / load muscles: When storing the muscle layout into a file, the coarse grid layout is stored together with the skin vertex indices that attach to each muscle. Keeping the vertex information speeds up and stabilizes the restoration of skin connections: it is made sure that a barely attached vertex does not lose its connection to a muscle due to limitations of numerical machine precision, for instance, when loading the face on another computer.

The speed of the muscle construction and insertion algorithms is essential for interactive work: when muscle shapes and connections change due to an editing operation, the result is immediately shown in the editor. In addition to a muscle's shape, information about the influenced mesh vertices, connections to other muscles, and attachment of muscles to skull or jaw are displayed, as shown in Figure 7.16.

7.6 Example: A Set of Human Facial Muscles

Using the muscle model and editing methods described in this chapter, I have created a set of the 24 major muscles in the human face responsible for facial expressions. The layout was created on the reference head geometry. Figure 7.17 illustrates the complete set. Table 7.1 gives details about the muscle representation.

The construction of most of these muscles is straightforward, the major exception being the orbicularis oris. This muscle is extremely flexible and needs to support many different mouth shapes for speech articulation. As can be seen in Figure 7.7, it is thus modeled as two parts representing the upper and lower lip, respectively. The lower lip part is attached to the jaw, rotating with it when the mouth opens. Both halves have their own axis of contraction, therefore allowing for independent contraction and protrusion settings.

The mylohyoideus muscle is special in that it was inserted only to give the model structural support in the area underneath the chin. Since there is no bone in this area, the skin would be only loosely flapping when the mouth is opened. It was not necessary, though, to build a complete platysma for this purpose.

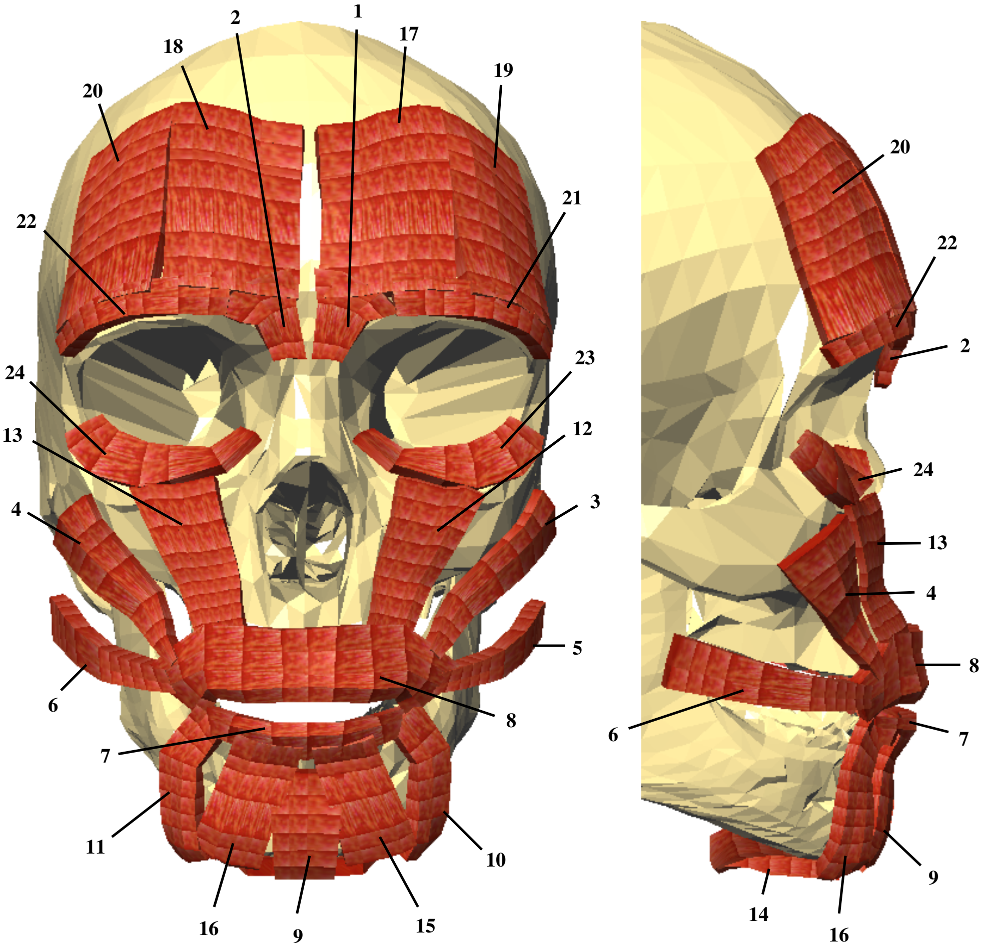


Figure 7.17: The muscle set on the reference head model, front and side views. The annotated numbers correspond to Table 7.1.

To get an estimate of the performance of the muscle deformation algorithms in the facial animation system, I have taken measurements from an animation that made use of all muscles, playing back at 50 fps on a 1.7 GHz Xeon CPU. The animation ran for a total of 5489 frames, causing 18573 muscle deformations. Figure 7.21 shows the different parts of the muscle update process in relation to the total frame time. As can be seen, the muscle update consumes only about 17% of the total available computation time at this update rate.

7.7 Discussion

The muscle model presented in this chapter extends current approaches, most notably LEE's piecewise linear model. The major advances lie in the more elaborate and flexible muscle geometry, the possibility to connect muscles to each other, and the increased flexibility in sphincter muscle contraction. All of these additional features have an impact onto the visual appearance of the face model.

The geometry of the muscles determines the attachments to the skin layer, and

Linear muscles

index	*name*	*description*	*att.*	*conn.*	*#p*	*#f*	*#s*
1,2	corrugator supercilii	causes wrinkling of eyebrows	s	17;21 18;22	6	3	5
3,4	zygomaticus major	lifts mouth angles	s	8	4	2	9
5,6	risorius	pulls mouth angles horizontally	s	8	4	2	12
12,13	levator labii	lifts lateral part of upper lip	s	8	5	5	10
17,18	occipitofrontalis (inner)	lifts inner part of eyebrows	s	1;21 2;22	6	5	8
19,20	occipitofrontalis (outer)	lifts outer part of eyebrows	s	21 22	6	5	8
14	mylohyoideus	structural support for chin	j	–	4	7	6
15,16	quadratus labii	pulls down lower lip	j	7	6	4	8
9	mentalis	pulls chin upwards	j	–	5	3	7
10,11	triangularis	pulls mouth angles downwards	j		6	2	8

Sphincter muscles

index	*name*	*description*	*att.*	*conn.*	*#p*	*#f*	*#s*
21,22	orbicularis oculi pars orbitalis	pulls down eyebrows	s	1;17;19 2;18;20	14	2	10
23,24	orbicularis oculi pars palpebralis	lifts lower eyelids	s		18	2	7
8	orbicularis oris (upper lip)	closes mouth, speech articulation	s	3;5;7;12 4;6;13	55	4	11
7	orbicularis oris (lower lip)	closes mouth, speech articulation	j	8;10;15 11;16	21	2	7

Table 7.1: Details of the muscle representation in the example set shown in Figure 7.17. The index column refers to that figure. The rightmost five columns are:

att.: muscle attachment; *j* for jaw, or *s* for skull

conn.: connections; indices of merged muscle(s) on left / right face halves

#p: number of points on control grids (excluding axis points for sphincters)

#f: the number of muscle fibers after construction from the initial grid

#s: the resulting number of control points per fiber after construction

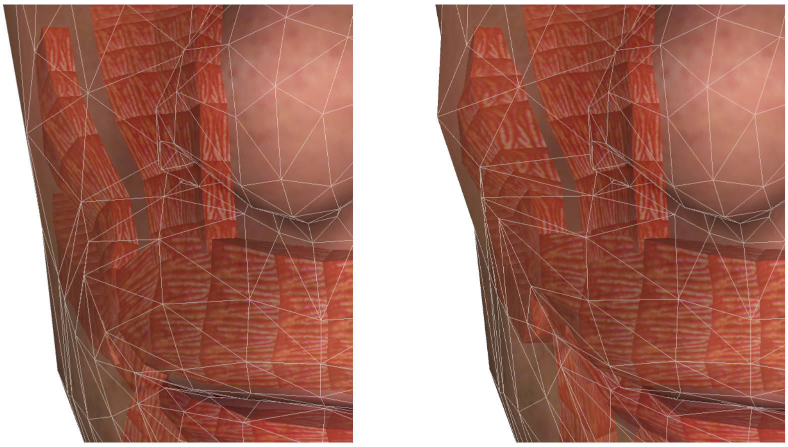


Figure 7.18: Bulging cheek during a smile

influences skin deformation when the muscle is contracted or stretched. The bulging heuristics used in the current implementation are very simple, but effective. It would be interesting to investigate other heuristics, for instance, preserving muscle volume [SPCM97, WV97]. I have experimented with using muscle bulging to simulate the effect of fat cushions in the face, for instance, on the cheeks: during a smile, there is a considerable buildup of tissue due to the contraction of the zygomaticus major. This can be reproduced partially with bulging, as illustrated in Figure 7.18. Still, while the skin clearly bulges out when the muscle is contracted, this effect should be much stronger. To this end, it would be necessary to represent the fat pads underneath the skin in that area either implicitly through another heuristics built into the muscle model or as independent deformable components.

The muscle connection constraints have shown to be important for animation pertaining to the mouth. Figure 7.19 demonstrates how the upper lip part of the orbicularis oris is raised when the zygomaticus major muscles raise the corners of the lip to form a

Figure 7.19: Connected muscles influencing each other during a smile. The orbicularis oris is pulled up by the zygomaticus major muscles, causing the muscles attached to the mouth angles to stretch.

Figure 7.20: Visemes rely heavily on the flexibility of the orbicularis oris model. Left to right: /f/, /i/, /u/. Note how upper and lower lip move independently and are able to not only contract, but also protrude or retract.

smile. As the mouth opens, the mouth angles are restrained by the connected upper lip parts, leading to elongation and straightening of the triangularis muscles. This results in a realistic indentation effect on both sides of the chin.

The two-part construction of the orbicularis oris muscle is essential for proper modeling of visemes, making speech animation much more realistic. Figure 7.20 shows a few examples for visemes.

Updating the whole muscle configuration is fast enough for a real-time animation system. While the computation time per frame is not negligible, as shown in Figure 7.21, it poses no bottleneck. The dominant part is the dynamic simulation used for maintaining muscle connectivity, which I nonetheless found indispensable to achieve plausible inter-muscle deformations.

Due to the automatic generation of muscle geometry from coarse layouts, a muscle can be automatically rebuilt from the grid for many different head models. This not only simplifies interactive muscle construction, but also makes it easy to adapt the

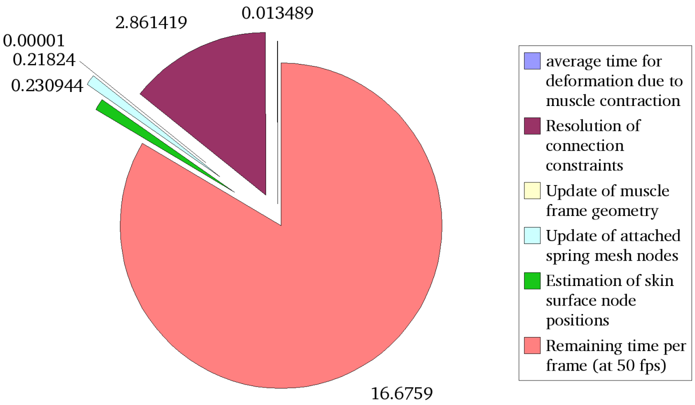


Figure 7.21: Timings for muscle updates. The values are given in ms, with a total frame time of 20 ms.

muscle layer to a deformed version of the reference head, as described in Chapter 8. In general, the simple fitting algorithm proposed here delivers good results. The control points in the originally specified grids are deliberately not changed by the algorithm to maintain a certain level of user control. This is mostly important for building the constraint groups, which connect nearby control points of muscles. Due to this restriction on control point placement, the level of refinement of the resulting muscle shape is sometimes quite high, especially in non-uniform, highly curved regions of the facial mesh such as the mouth angles. The recursive subdivision is stopped when the refined segments become too small, which can lead to leaving the allowed skin-muscle distance range. Also, the refinement method doesn't consider the actual segment shape, which becomes more apparent with the use of ellipsoidal shapes than with box shapes: a curved region of the mesh could be approximated with a single ellipsoid, instead of subdividing the muscle into many small segments.

The interactive muscle editing tool has been tested on human and non-human head models obtained from range scans (see also Figure 10.1 in Chapter 10). A human face muscle set has been designed with the editor on the reference head model based on anatomical illustrations of the facial musculature. Experimentation was necessary to find out which muscles are essential to achieve full-range expressiveness of the face. A fair amount of fine tuning had to be performed to get satisfactory muscle connections and influence on mesh vertices without undesirable artifacts. For instance, attaching muscles to the vertices of a coarse triangle mesh can be a delicate task, because the attachment is based on a strict distance threshold to the muscle surface. Editing a muscle can easily lead to small changes of the muscle surface that change the attachments unintentionally. Even a single vertex that has not been properly attached can have drastic effects on the animation. The visual feedback given in the editor has shown to be very important and helpful here. Updating the muscle shapes after changing the layout typically takes only a fraction of a second, thus allowing interactive re-editing of the muscle layout with immediate feedback on the resulting muscle shape and attachments. In summary, the editor simplifies the process of creating a set of facial muscles, and gives great flexibility to model different anatomical patterns. Due to the complexity of the resulting muscle network, the final tweaks still require a considerable amount of time. It is thus highly desirable to have a muscle layout automatically transferred to new models, which is shown in Chapter 8.

CHAPTER 8

Making New Faces

> *God hath given you one face, and you make yourselves another.*
> *– William Shakespeare*

In the preceding chapters I have described the structure of the virtual head model and how it is animated. During construction of the model, a variety of interactive tools are employed, both commercial and self-written. From the preparation of the head geometry to the layout of the muscles a sizable amount of interactive work is required. For a system that strives to be useful in practice, this workload must be reduced. Since human head models share the basic geometric characteristics and anatomical structure, it is in fact possible to derive new models from one prototype in a largely automated fashion.

In this chapter, I propose techniques that achieve this goal by deformation of all parts of an existing structured head model, adapting it to individual target geometry. The deformation is specified by a small number of landmarks (feature points) defined on the skin and skull layers of the model. By changing the location of the features and the relations between them, the geometry of skin, muscles, and skull can be flexibly edited. The approach is not restricted to human heads, but easily carries over to other virtual creatures, such as real or fantasy animals, as long as they can be modeled by anatomical means with a skull, muscles, and skin. The structured models for the variety of human faces appearing in this dissertation were created using these techniques.

The variability in the shapes of the human head and face is traditionally studied in the field of anthropometry. The literature is rich with insights about the differences in facial proportions among and between various human populations. Extensive statistical data that has been collected over the course of decades is put to use in the methods presented in this chapter.

In the following section, I give a review of existing work on adaptation of virtual head models, before moving on to a brief introduction to anthropometry and the related field of morphometrics. The structural head deformation technique itself is presented in Section 8.3. The usefulness of the method is illustrated thereafter by a number of applications in Sections 8.4 to 8.6, where the anthropometric knowledge built into the technique is put to use. These applications are:

Creation of virtual heads from real faces: The reference head model is deformed to match geometry obtained, for instance, from a range scanner. Extensive preparation of the scanned geometry for animation purposes thus becomes unnecessary.

Growth simulation: By redefining the relations between landmarks on the skin according to age-related statistics, facial proportions of a virtual head can be changed, resulting in a plausible model of the same person at a different age.

Face reconstruction: The structural deformation is usually driven by the target skin geometry. If instead the target is a skull model, muscle and skin layers can be created on top using statistical data, resulting in an animatable face.

A discussion of the overall approach concludes this chapter.

8.1 Related Work on Head Deformation

Many facial modeling and animation systems make use of a generic head model (see also Chapter 3), which is often associated with a deformation method suited to the respective application. The approaches discussed over the next paragraphs can be divided roughly into *surface-based* and *volume-based* deformations.

8.1.1 Surface-Based Deformation

The adaptation of a head model to scan data is performed in two dimensions by WATERS *et al.* [WT91] and LEE *et al.* [LTW95]. The vertices of a planar projection of the generic head mesh are moved in the plane, aligning them to facial features and increasing the density of the mesh in the more articulate face areas. This method is designed specifically for matching a generic mesh to geometry that is parameterized over the plane, as is the case for cylindrical range scans. Facial muscle attachments are stored relative to the generic mesh and can thus be adapted automatically, as LEE points out. A planar version of the generic mesh has to be generated from a three-dimensional generic mesh first, which can be difficult to impossible, depending on the geometry and topology of the mesh.

The approach taken by DECARLO *et al.* [DMS98] is based on variational modeling [WW92], operating directly on a B-spline surface in three-dimensional space. Variations of a generic head model tagged with landmarks are generated by altering several linear and non-linear constraints that are formulated in terms of surface properties, for instance, landmark distances or tangent directions. Most of the constraints are not directly applicable to polygonal geometry. The method does not include any structure inside the head model, though the authors have proposed this for future extensions of the method. Most notably, the technique demonstrates the practical use of anthropometric data for creating new head models with realistic proportions.

8.1.2 Volume-Based Deformation

For transformation of a three-dimensional structure in space, it seems more appropriate to define a deformation of a volume rather than a surface. In SEDERBERG's and PERRY's free form deformation (FFD) approach, a control lattice is used to define a mapping from undistorted to distorted space [SP86]. Many variations of the original formulation exist, in particular to hide the control lattice from the user [HHK92, LWCS96]. In face modeling, ESCHER *et al.* [EPMT98] use Dirichlet free form deformation (DFFD [MMT97]) for generation of custom heads from a generic polygonal model by defining new positions for facial feature points.

Volume morphing using radial basis functions (RBF) is another popular deformation method that has been used for facial modeling. PIGHIN *et al.* [PHL$^+$98] deform a generic mesh surface according to feature points defined on the mesh and corresponding target points derived from a number of photographs. For an accurate match to the pictures, the specification of more than a hundred feature points is considered to be necessary. NOH *et al.* [NN01] adapt one face mesh to another face mesh for the purpose of transferring animations. They note that the match computed with the radial basis function method is not as accurate as desirable when the set of feature points is sparse. The problem is solved in their method by cylindrically projecting the deformed mesh onto the target after performing the initial RBF fitting.

Deformation of a generic model including muscles has been demonstrated by AOKI *et al.* [AHTN01]. Adaptation to a target head model is performed by fitting the generic model in separate stages by width, height, and depth, using simple geometric transformations. The method uses X-ray images to capture front and side views of the head structure, relying on accurate alignment of the head when these images are taken. Muscles are represented by line segments, connecting the skin and skull layers. Neither the muscle model, nor their handling in the deformation of the generic model is discussed in detail.

8.1.3 Summary

To conclude this small survey, existing head model deformation methods deal mostly with skin geometry, not with underlying structure. The few approaches that include anatomical structure do not show the flexibility needed to handle, for instance, variations in muscle or skin thickness. Still, it can be seen that volume morphing is appropriate for handling a structured model since the deformation function can be applied to the whole head volume. Nearly all discussed methods rely on the use of feature points, which are specified either manually [DMS98, PHL$^+$98, EPMT98] or using simple heuristics [NN01]. The main tradeoff lies in the sparseness of the feature point set versus precision of local deformation: a conveniently parsimonious set of feature points does not allow detail changes in the deformed geometry.

Remarkably, the feature points used in computer graphics head modeling systems correspond conceptually to the landmarks used in anthropometry for obtaining physical facial measurements. Thus, a deformation technique based on appropriately chosen feature points suggests itself since this allows the direct use of anthropometric data. In the following, the field of anthropometry in general and the use of facial landmarks in particular is introduced.

8.2 Measurements & Statistics: Face Anthropometry

Anthropometry is a branch of medical research that deals with measuring and analyzing absolute quantities and proportions of the human body, including the head. Face anthropometry has applications in clinical practice as well as in other fields.

Clinical diagnostics: The data collected from a population of individuals can be analyzed to extract a norm for facial features. Comparison with patient data can help to indicate the existence of deformities, possibly leading to discovery of an illness.

Operation planning: If size and shape of a deformity are quantifiable, one can make more exact statements about necessary corrections. Data about growth-related changes in the face helps to predict the effect of an operation over several years. For cosmetic surgery, some studies examine the aesthetics and attractiveness of human faces, attempting to find generally applicable measurements and proportions. Questionable as this may sometimes seem, anthropometric measurements still allow for a common language for exact description of facial shape and proportions.

Forensics: An important area of work for the forensic pathologist is the identification of a crime victim from his remains, often merely a skeleton. Examination of anthropometric measurements on the skull helps to find out more about age, sex, and ethnic origin of a person. If a photograph is available, a comparison can confirm or reject a hypothesis of the victim's identity. Reconstruction of complete faces from a skull is also made possible through knowledge about the relation of skin and skull. Police forces use artificial aging of missing children to increase the chances of finding them after years: statistical changes due to growth are applied to a picture of the child, approximating his or her current look.

The Arts: Anthropometric knowledge influences the design of portraits and sculptures. This includes applications in computer graphics, such as automatic generation of plausible human head geometry using anthropometric measurements [DMS98].

Traditionally, facial measurements are taken by a medical doctor using an assortment of measuring devices such as rulers, tapes, and goniometers. Some measurements are of a rather qualitative nature, i.e., hard to define in numbers, such as "the shape of the face is concave". These judgments depend to a large degree on the experience of the examiner. For precise measurements, a standardized set of landmarks is defined on the head. Landmarks are prominent feature points such as "the tip of the nose", which can be found by visual examination or palpation. The subject's head is typically aligned to a standard position, the *Frankfort horizontal*, often also termed *FH position*. With this alignment, the most important types of measurements are distances in the vertical, horizontal, and depth dimensions, as well as angles between three landmarks. Figure 8.1 shows a standard set of landmarks on the face, and a typical measurement of the head size.

8.2.1 Morphometrics

A branch of statistics studying statistical measurements of biological shapes has only recently become a discipline in its own right, namely *morphometrics*. Building on multivariate statistics, the influential work of BOOKSTEIN [Boo97a, Boo97b] combines geometry, biology, and statistics to a landmark-based approach for analysis of

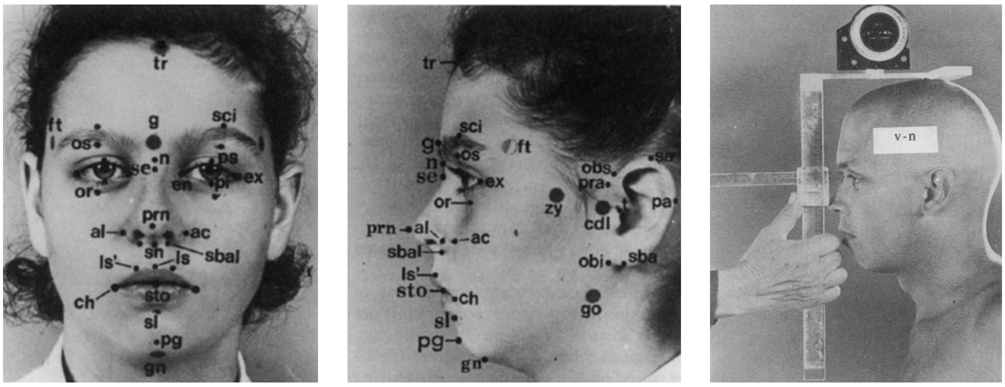


Figure 8.1: From left to right: landmarks on a human face, front and side views; physical measurement of the anterior height of the head. Images: [Far94]

shapes and shape deformations. Landmarks, in this context, are point samples on the surface of a biological shape under examination. A measure of similarity between two related shapes can be established by looking at the type and degree of deformation that maps landmarks defined on one shape to their counterparts on the other shape. Such similarity usually indicates that some biological process has moved the landmarks, or deformed the shape. To describe this deformation across the whole surface or volume, instead of only at the sample points, a suitable, “natural” deformation function has to be found. BOOKSTEIN proposes thin-plate splines for this purpose: the bending energy of the surface undergoing deformation is minimized, which is deemed a plausible model for biological shape changes.

BOOKSTEIN already foresaw the importance of computer generated animations in his work when he envisioned how modifications of biological shapes can be performed and understood in real time on the screen. His findings motivate the deformation method described here.

8.3 Landmark-Based Head Deformation

The thin-plate spline based approach advocated by BOOKSTEIN for two-dimensional data can be easily translated to the three-dimensional setting [Boo97b]. This section gives details on the practical construction of the deformation function for a source and a target set of landmarks. This mapping specifies a source object space deformation that is used to warp the complete head structure to the target configuration.

8.3.1 Setting up the Warp Function

The problem can be stated as one of interpolation: let $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{q}_i \in \mathbb{R}^3$, $i = 1, \ldots, N$, be two sets of N landmarks. The $\mathbf{p}_i$ lie on the geometry to be deformed (the *source landmarks*), and the $\mathbf{q}_i$ correspond to the features on the target head (the *target landmarks*). A function $\mathbf{f} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is needed that maps the $\mathbf{p}_i$ to the $\mathbf{q}_i$:

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

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:

$$\mathbf{f}(\mathbf{p}) = \sum_{i=1}^{N} \mathbf{c}_i \phi_i(\mathbf{p}) + \mathbf{R}\mathbf{p} + \mathbf{t}, \tag{8.1}$$

where $\mathbf{p} \in \mathbb{R}^3$ is a point in the volume, $\mathbf{c}_i \in \mathbb{R}^3$ are (unknown) weights, $\mathbf{R} \in \mathbb{R}^{3\times3}$ adds rotation, skew, and scaling, and $\mathbf{t} \in \mathbb{R}^3$ is a translation component. Following BOOKSTEIN, I simply use $\phi_i(\mathbf{p}) := \|\mathbf{p} - \mathbf{p}_i\|_2$, which minimizes bending energy for the deformation [Duc77]. The additional constraints

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

are given to remove affine contributions from the basic function term [PHL$^+$98, CBC$^+$01]. By setting up a system of linear equations relating source and target landmarks, the unknowns $\mathbf{R}$, $\mathbf{t}$, and $\mathbf{c}_i$ can be found simultaneously. Three matrices are constructed:

$$\begin{aligned}
\mathbf{B} &= \begin{pmatrix} \mathbf{q}_1 & \dots & \mathbf{q}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}^T \in \mathbb{R}^{(N+4)\times3}, \\
\mathbf{P} &= \begin{pmatrix} \phi_1(\mathbf{p}_1) & \dots & \phi_N(\mathbf{p}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{p}_N) & \dots & \phi_N(\mathbf{p}_N) \end{pmatrix} \in \mathbb{R}^{N\times N}, \\
\mathbf{Q} &= \begin{pmatrix} \mathbf{p}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{p}_N^T & 1 \end{pmatrix} \in \mathbb{R}^{N\times4}.
\end{aligned}$$

Now a linear equation system of the form $\mathbf{AX} = \mathbf{B}$ is set up with

$$\begin{aligned}
\mathbf{A} &= \begin{pmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{Q}^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(N+4)\times(N+4)}, \\
\mathbf{X} &= \begin{pmatrix} \mathbf{c}_1 & \dots & \mathbf{c}_N & \mathbf{R} & \mathbf{t} \end{pmatrix}^T \in \mathbb{R}^{(N+4)\times3}.
\end{aligned}$$

This linear system is solved using a standard LU decomposition with pivoting. A point $\mathbf{p} \in \mathbb{R}^3$ can now be transformed according to Equation (8.1).

8.3.2 Deforming the Head Structure

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

1. The *skin mesh* is deformed by direct application of the function to the vertices of the mesh (Figure 8.2(a)).

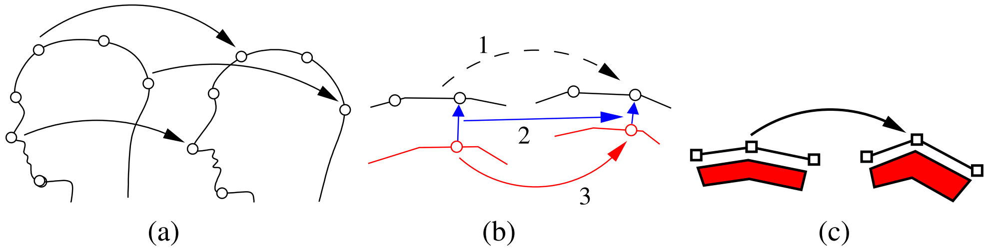


Figure 8.2: Mapping the head structure to target geometry. a) Warping the skin mesh: source skin landmarks are mapped directly to target skin landmarks. b) Warping the skull mesh: after mapping the skin (step 1), new skin / skull offset vectors are computed (step 2); the vector endpoints define source and target skull landmark positions, used to set up a mapping for the skull geometry (step 3). c) Muscle warping: the skin deformation is applied to the muscle grids and the muscle geometry is rebuilt afterwards.

2. The landmarks on the *skull mesh* are related to their counterparts on the skin by an offset vector from each skull landmark to the corresponding skin landmark (Figure 5.1(d)). The deformation function for the skull is obtained by displacing the target skin landmarks along the negated offset 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 (Figure 8.2(b)). In this approach, adjustments of local skin thickness amount to a simple change of scale for the offset vectors.

3. *Muscles* are specified by their grid, painted onto the skin. To transfer the muscles to the new geometry, I apply the deformation function to the grid vertices and re-compute the shape, as described in Section 7.4 (Figure 8.2(c)). This rebuild process also adds flexibility to adapt the muscle geometry, for instance, to changes in skin thickness.

4. For the other facial components eyes, teeth, and tongue, only position and scale can be updated automatically, due to their representation as rigid pieces of geometry. 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.

8.4 Application 1: Head Models from Range Scans

A primary task in facial modeling is the reproduction of heads of real individuals. One way to acquire the geometry of a person's 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. Hair-covered regions, for instance, are problematic for some scanners. This often requires extensive post-processing of the mesh.

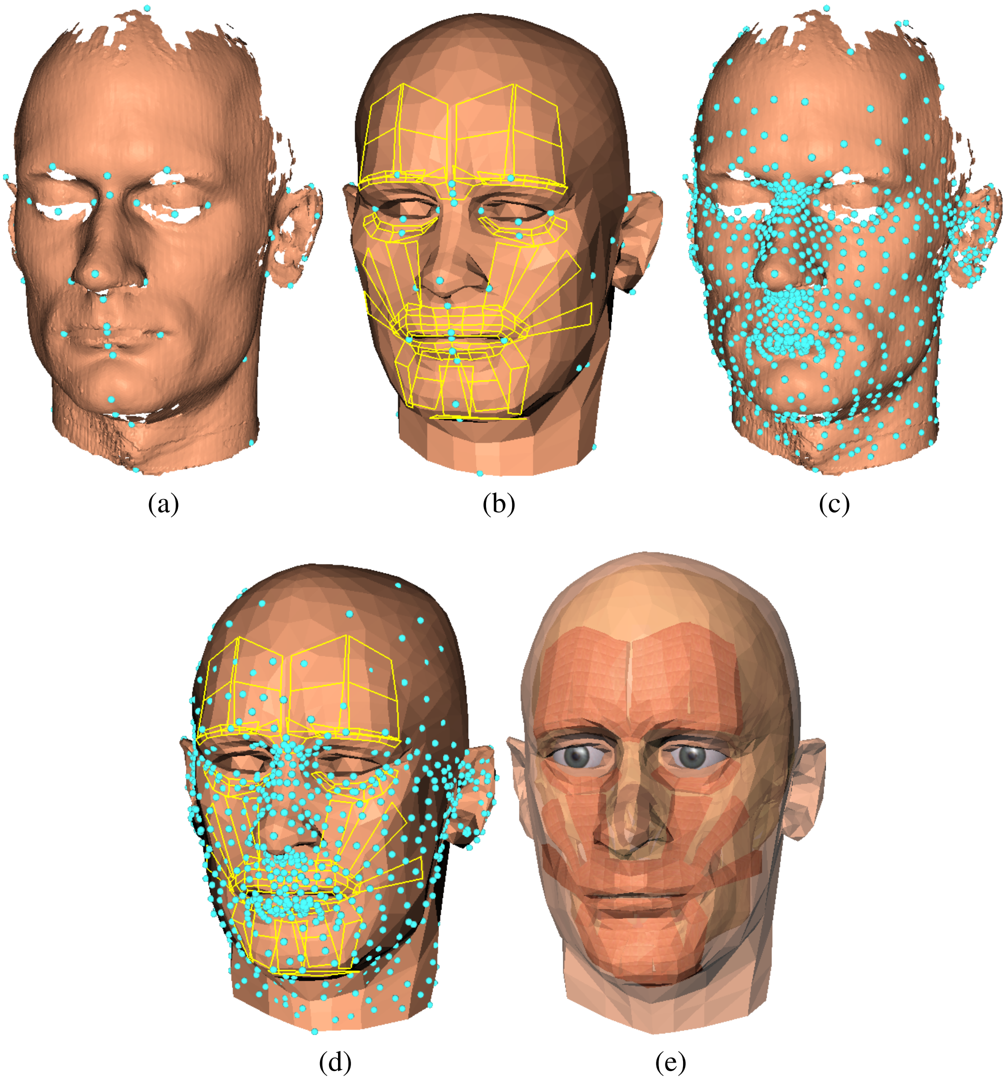
- The geometry is heavily oversampled: direct conversion to a triangle mesh regularly yields hundreds of thousands of polygons. For real-time physics-based animation, the complexity needs to be reduced by several orders of magnitude to about one to five thousand polygons. Available mesh simplification techniques [CMS98] unfortunately don't give enough control over the mesh connectivity to generate satisfyingly animatable models. Edges must be aligned to facial features, and the mesh structure should reflect the basic symmetry of the face. Also, flat areas in the scanned geometry cannot blindly be simplified: they don't necessarily remain flat in animation, requiring higher local mesh resolution for proper deformation.
- Some relevant parts cannot be scanned, such as the inner part of the lips.

To avoid these problems, a generic head model can be adapted to scanned target geometry. I employ the landmark-based deformation approach to create a fully animatable, individual model from a reference head model and a range scan. No parameterization of the scan data or the reference head mesh is needed. The method is thus not restricted to scans from cylindrical range scanners [LTW95], but can directly process data from arbitrary sources. Concerning this, there are 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 demonstrated in Section 8.5.

8.4.1 Adaptation Process Overview

An individual head model is created from range scans of its real-world counterpart. The models depicted in this dissertation, for instance, were created with a structured light scanner. Using the associated software package, a high resolution triangle mesh representing the skin geometry was generated from each data set. No further fixes such as hole filling need to be applied to the scans. This high resolution mesh is called 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. The procedure is as follows (see also Figure 8.3):

1. $\mathcal{M}^*$ is tagged with a set of landmarks $\mathcal{L}^*$ corresponding to the set $\mathcal{L}$ defined on $\mathcal{M}$. This is an 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.
7. The skull is warped once more to finally adjust the skull / skin relationship, as described in Section 8.3.2.



(a) (b) (c)

(d) (e)

Figure 8.3: Adaptation of the reference mesh to scan data using feature mesh refinement. a) Initial defective target mesh $\mathcal{M}^*$ from range scans with landmarks added; b) source mesh $\mathcal{M}$ and muscle outlines after first deformation; c) landmarks on target geometry after three refinements; d) deformed source mesh after three refinements and warps; e) final mesh geometry with adapted skull and muscles.

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 7, the given target skull / skin distance is enforced at the sparse locations where landmarks on skin and skull are paired.

The specification of the landmarks $\mathcal{L}^*$ and the landmark set refinement procedures are discussed in detail in Sections 8.4.2 and 8.4.3.

8.4.2 Specifying Landmarks

Only a sparse set of landmarks needs to be specified 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 [PHL+98]. Since the landmarks are taken from the standard set described in the anthropometric literature, they are well-defined and easy to identify. To further ease the repeated point-and-click task of landmark placement, I again make use of the deformation function: given three or more landmarks specified manually, a mapping can already be set up from the source set of landmarks $\mathcal{L}$ to the target set $\mathcal{L}^*$. All landmarks are then copied and warped from $\mathcal{L}$ using this function, resulting in a rough approximation of the desired landmark distribution, as shown in Figure 8.4. Through manual inspection and correction more landmarks are repositioned and fixed in their final 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 in space.

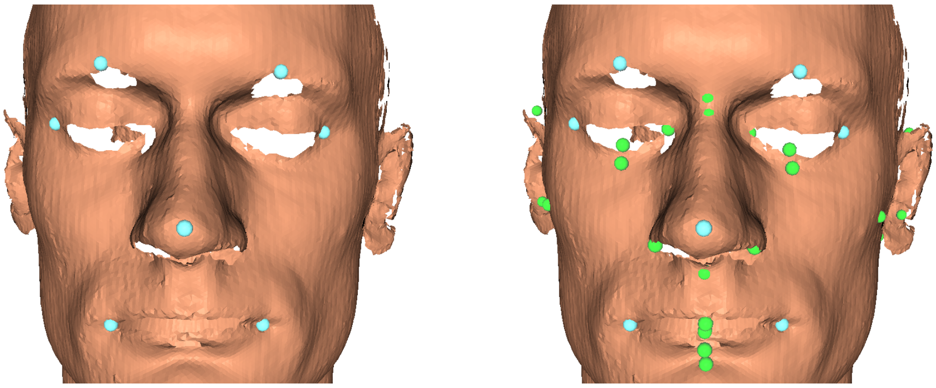


Figure 8.4: Interactive specification of landmarks. Left: A few landmarks (cyan) are specified manually. Right: All other landmarks (green) are copied from the reference model using the warp set up by the initial landmark set.

8.4.3 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 8.3(b). But, since the landmark distribution is very sparse, the details in the facial geometry of $\mathcal{M}^*$ are usually not well captured.

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

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

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

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

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

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

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

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

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

Table 8.1: The refinement algorithm for landmark sets on source and target geometries; see text for detailed explanation.

The user should not be burdened with specification of hundreds of feature points, so I have developed an automatic procedure that refines the landmark sets $\mathcal{L}$ and $\mathcal{L}^*$ to achieve a better match. The algorithm outline is shown in Table 8.1.

To be able to refine the standard set of landmarks automatically, the landmarks are interpreted as the vertices of a special triangle mesh, called the *feature mesh* in the following. Figure 8.5 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 care has to be taken 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 $\mathcal{B}$, and the corresponding scalar displacements along the normal vector are stored in another set $\mathcal{D}$. The right half of

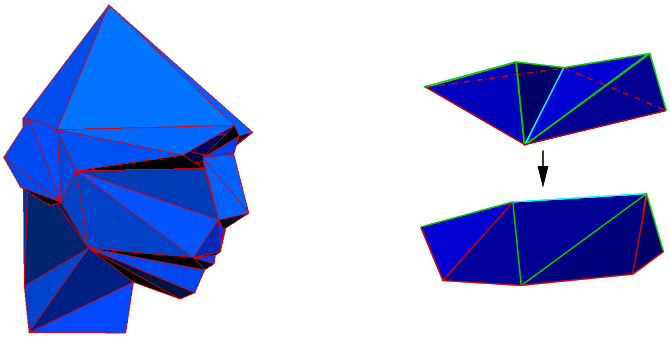


Figure 8.5: Left: A feature mesh is constructed by connecting the landmarks on the head geometry, forming a triangle mesh. Right: Flipping edges after subdividing the mesh improves surface smoothness: new vertices on two neighboring subdivided triangles are connected by the cyan edge previously connecting the original triangles. Red edges have also been flipped.

Figure 8.6 outlines the subdivision and projection steps for one triangle of $\mathcal{M}^*$.

Now, counterparts for the newly created landmarks need to be found on $\mathcal{M}$. Since $\mathcal{F}$ and $\mathcal{F}^*$ have the same structure, and the geometries are already in good alignment, the same refinement is repeated on $\mathcal{F}$, using the information from $\mathcal{B}$ and $\mathcal{D}$. 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 $\mathcal{L}^*$, the algorithm finds the nearest intersection along a ray starting at the vertex, corrects its position and creates a landmark at that point, adding it to $\mathcal{L}$. The left half of Figure 8.6 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 quality of the feature mesh surfaces, as illustrated in Figure 8.5. 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 local sampling of the surface. Also, the triangle normals are filtered once by averaging with their neighbors.

Using the two refined versions of $\mathcal{L}$ and $\mathcal{L}^*$, a new deformation function is constructed. 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, quickly stabilizing the deformed mesh to a good fit of the target geometry. In practice, after only three iterations of the refinement procedure the geometry will have adapted optimally. At this level of refinement, the precision of the match is already limited by the discretization of the source mesh, which is much coarser than the scan data.

Since a triangle mesh generated directly from range scan data is used as the target geometry, there are usually large areas of the head geometry where there is no data, often interspersed with small specks of samples (for instance, at the back of the head, see Figure 8.3(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 heuristics based on surface normals and landmark / surface distance are employed to find and

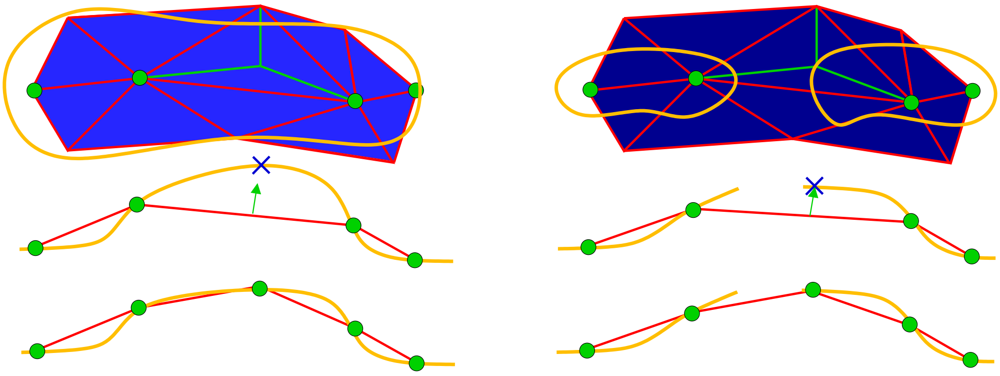


Figure 8.6: Refining corresponding triangles in the source (left, light blue) and target (right, dark blue) feature meshes. Top: The mesh is typically well-behaved in the source geometry, and fragmented in the target (orange curves). Where vertices of the target feature mesh project onto the geometry, 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.

rank the acceptable sites or reject creation of a landmark. The ray intersection before repositioning a new vertex in the source feature mesh is much less critical since it is operating on the deformed reference mesh, which has perfectly well-behaved geometry. Here, the algorithm just has to make sure not to intersect with backfacing parts of the geometry.

8.4.4 Discussion

The landmark-based fitting has shown to work robustly on raw meshes obtained from range scans of a variety of individuals. Even though automated processing is always a desirable goal, interaction can be advantageous when landmarks have to be specified: the initial parsimonious landmark set often has to be created on incomplete data, where large parts of the geometry are missing, and cannot be estimated automatically in a reliable way. It is thus necessary to place landmarks “in empty space”, guided by an estimate of the landmark distribution as described in Section 8.4.2. Simple heuristics [NN01] rely on well-behaved mesh geometry and are not suitable for most of the anthropometric standard landmarks. The refinement and fitting stages of the adaptation process are aware of missing surface parts, leading to robust adaptation. Mesh fixing and hole filling become abdicable, procedures that can otherwise easily take several hours of interactive work.

In practice, specifying the reference set of 60 landmarks takes only about 10 to 20 minutes, depending on the quality of the scan data. While the specific set of anthropometric landmarks is useful for further measurements and deformation of the head model (see next section), the adaptation to scan data can in principle be performed on an arbitrary set of landmarks. The computational cost of the landmark refinement algorithm largely depends on ray / mesh intersection tests and point / triangle distance computations. A great speed-up can be expected from optimization of these tests. In the current implementation, the run time totals around 5 minutes for the fitting process on a 1 GHz PC with a 100k triangle target mesh. Starting with given scanned geometry, the whole process of creating an animatable head model including the tuning of eye, teeth, and tongue positions takes approximately 30 minutes, not including generation of the skin texture.

8.5 Application 2: Growth and Aging

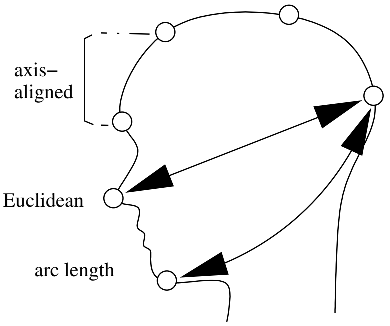
A challenging problem in facial modeling and animation is the simulation of growth and aging. Besides the Arts, important applications exist in the forensic sciences and in medicine: how does the child that went missing ten years ago look now? What are the long-term effects of a rhinoplastic operation? Typically, a skilled artist is needed to infer age-related changes from photographs, relying on anthropometric knowledge and data [Tay01]. Computer-aided methods are cautiously beginning to be used in forensic applications, based on two-dimensional image processing [Tay01, Chapter 7][Rus92, All90].

8.5.1 Related Work

In computer graphics, research on aging in human faces has so far concentrated on the appearance of the skin, for the most part neglecting the considerable geometric changes that occur during growth. WU *et al.* [WKMMT99, WMTT94] focus on generation of expressive wrinkles and skin aging effects. Their muscle-driven face model incorporates viscoelastic properties of the skin. LEE *et al.* [LWMT99] 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.* [LTC99] present a statistical face model to isolate age variations in face images for age estimation and aging simulation. TIDDEMAN *et al.* [TBP01] 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.

8.5.2 Head Shape Changes during Growth

Drastic changes in head size and facial proportions occur between infancy and maturity, where the most actively growing region of the head is the face. While TODD *et al.* [TLSP80, PW96] have devised a simple mathematical model for the changes of the face up to adulthood, growth patterns are in general highly irregular in time as well as in the regions affected. Several studies of head growth during childhood find separate rhythms for the development in facial height, depth, and width, as well as alternating phases of accelerated and slowed-down growth [FPH92, EN72, Tan62, GSD90]. There are also significant differences in development between males and females, for instance, the mandible becomes much more pronounced during male growth. Most



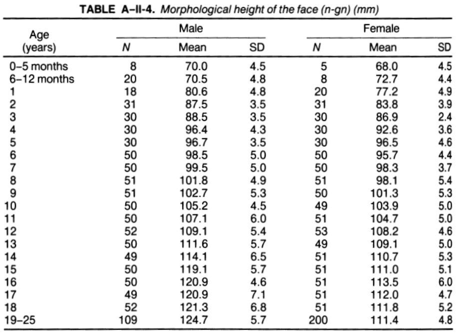


TABLE A–II-4. *Morphological height of the face (n-gn) (mm)*

Age (years)	Male			Female		
	N	Mean	SD	*N*	Mean	SD
0–5 months	8	70.0	4.5	5	68.0	4.5
6–12 months	20	70.5	4.8	8	72.7	4.4
1	18	80.6	4.8	20	77.2	4.9
2	31	87.5	3.5	31	83.8	3.9
3	30	88.5	3.5	30	86.9	2.4
4	30	96.4	4.3	30	92.6	3.6
5	30	96.7	3.5	30	96.5	4.6
6	50	98.5	5.0	50	95.7	4.4
7	50	99.5	5.0	50	98.3	3.7
8	51	101.8	4.9	51	98.1	5.4
9	51	102.7	5.3	50	101.3	5.3
10	50	105.2	4.5	49	103.9	5.0
11	50	107.1	6.0	51	104.7	5.0
12	52	109.1	5.4	53	108.2	4.6
13	50	111.6	5.7	49	109.1	5.0
14	49	114.1	6.5	51	110.7	5.3
15	50	119.1	5.7	51	111.0	5.1
16	50	120.9	4.6	51	113.5	6.0
17	49	120.9	7.1	51	112.0	4.7
18	52	121.3	6.8	51	111.8	5.2
19–25	109	124.7	5.7	200	111.4	4.8

Figure 8.7: Left: different types of landmark measurements: aligned to the vertical axis, Euclidean, and arc length; right: table of age-related changes for one specific distance measurement. Table: [Far94]

studies only consider ages up to maturity at age 18. At this point, head growth is considered to be settled. Some publications indicate ongoing changes in post-adolescent years, though: VERCAUTEREN [Ver90] found "statistically significant" differences between age groups in a study of 25–54 year old Belgians, such as a decrease in head width. TANNER [Tan88] states that cephalic and facial dimensions increase continuously throughout the life span. After age 20, only minor changes in dimensions of 2 to 4 percent can be observed, though.

The analysis of the anatomical details studied in anthropometric research is beyond the scope here, but it can nonetheless be asserted that the growth of the human head is a complex process that cannot be easily cast into a simple model. The most interesting and drastic changes in head shape occur during childhood, and statistically relevant material has been collected almost exclusively for that age period. The remainder of this section demonstrates the use of anthropometric data for simulating head growth during childhood and adolescence, employing the landmark-based head deformation method.

8.5.3 Tabulated Growth Data

For the methods described here, I make use of tabulated measurements collected by FARKAS [Far94] for a sample of the North American population. The data consists of distance relations for pairs of landmarks on male and female faces of the Caucasian type, spanning the ages of 1 year up to 25 years. One or more of three different types of distance measurements are given for each landmark pair:

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

Each such statistical measurement is given by its mean value μ and standard deviation σ. Figure 8.7 illustrates the different types of measurements, and shows sample table data.

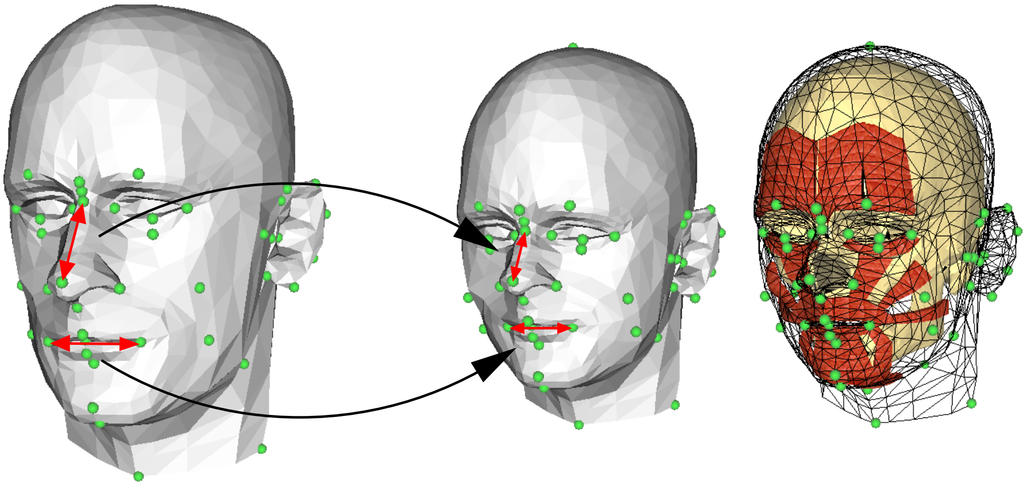


Figure 8.8: Deriving new measurements from landmark pairs on an individual head model. Left: male model of 31 years. Middle: constraint system solved for target age of 5 years. Right: transformed head structure.

8.5.4 Age Changes on the Virtual Head Model

The basic procedure for changing the age of a given individual head model is depicted in Figure 8.8: the set of landmarks on the model is used to obtain measurements that can be compared to the tabulated statistical data for the modeled person's age and sex. These measurements are then updated according to a user-specified change in age by a constraint resolution mechanism, resulting in another set of landmarks corresponding to the target age. The mapping from the source set of landmarks to the target set defines the deformation of the head model. From the available textbook data, 39 axis-aligned distance measurements, 25 Euclidean distances, and 6 arc lengths are used to define the system of constraints, using the landmark set shown in Figure 8.9. These measurements are summarized in Table 8.2.

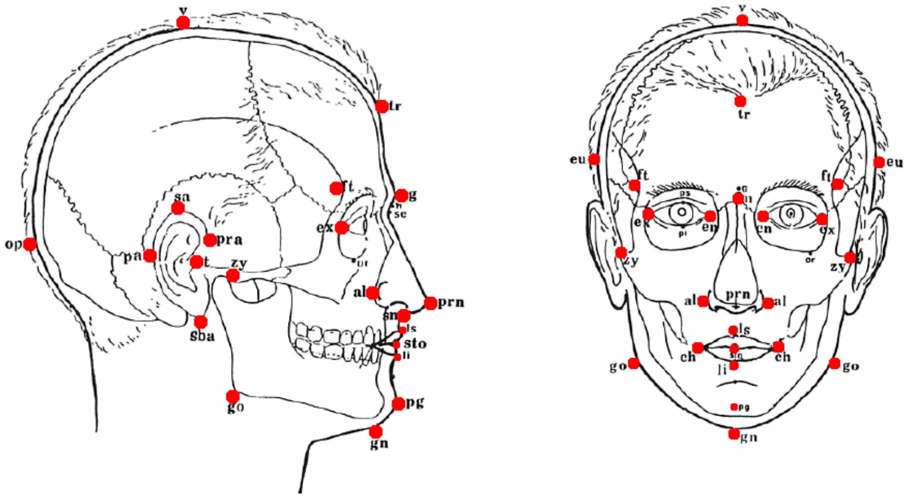


Figure 8.9: Landmark set used for age-related changes. Images after [Far94]

	Vertical	*Horizontal*	*Depth*	*Euclidean*	*Arc length*
Head	v-n v-gn v-po v-t	t-t	g-op		
Face	n-gn n-sto sn-gn sto-gn sl-gn	zy-zy go-go		t-g n-t sn-t gn-t go-gn	sn-t gn-t
Orbits	or-sci	en-en ex-ex ex-en	en-se	en-se ex-obs ex-t ex-go	
Nose	n-sn sn-sto	al-al	sn-prn en-se	n-prn	
Lips and mouth	sn-sto sn-ls ls-sto sto-li li-sl sto-sl	ch-ch		ch-t	ch-t
Ears	sa-sba obs-obi		pra-pa op-po	n-obs	

Table 8.2: Distance measurements used for age changes; see Figure 8.9 for landmark sites. Some landmark pairs are duplicated for the left and right halves of the model.

The virtual head model is placed in FH (Frankfort Horizontal) position, aligning the horizontal, vertical, and depth dimensions used for the measurements to the x, y, and z axes in the object's local coordinate system. The current value d_c for a distance measurement is first computed directly from the landmarks on the model. Given age and sex for the head 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 target age, the final distance value d_t at the same relative position in the distribution is computed:

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

where σ_c and σ_t are standard deviations for the current and target age, respectively. Mapping the deviation from the statistical mean in this way helps to retain the indi-

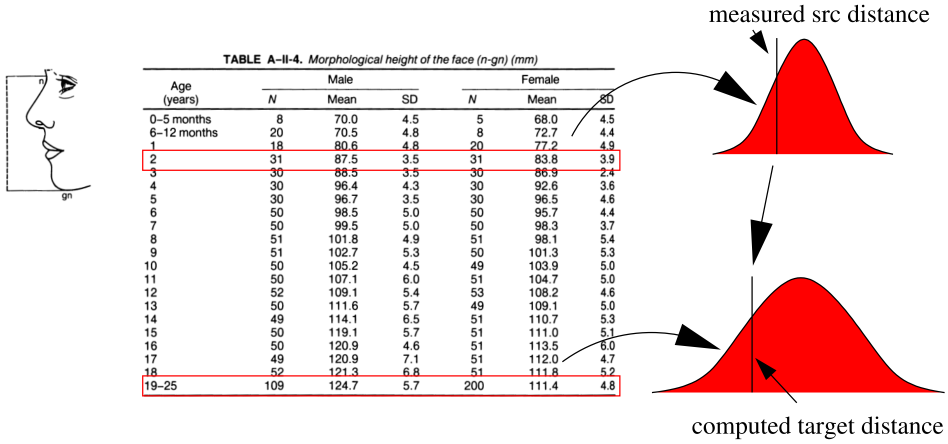


TABLE A–II-4. *Morphological height of the face (n-gn) (mm)*

Age (years)	Male			Female		
	N	Mean	SD	*N*	Mean	SD
0–5 months	8	70.0	4.5	5	68.0	4.5
6–12 months	20	70.5	4.8	8	72.7	4.4
1	18	80.6	4.8	20	77.2	4.9
2	31	87.5	3.5	31	83.8	3.9
3	30	88.5	3.5	30	86.9	2.4
4	30	96.4	4.3	30	92.6	3.6
5	30	96.7	3.5	30	96.5	4.6
6	50	98.5	5.0	50	95.7	4.4
7	50	99.5	5.0	50	98.3	3.7
8	51	101.8	4.9	51	98.1	5.4
9	51	102.7	5.3	50	101.3	5.3
10	50	105.2	4.5	49	103.9	5.0
11	50	107.1	6.0	51	104.7	5.0
12	52	109.1	5.4	53	108.2	4.6
13	50	111.6	5.7	49	109.1	5.0
14	49	114.1	6.5	51	110.7	5.3
15	50	119.1	5.7	51	111.0	5.1
16	50	120.9	4.6	51	113.5	6.0
17	49	120.9	7.1	51	112.0	4.7
18	52	121.3	6.8	51	111.8	5.2
19–25	109	124.7	5.7	200	111.4	4.8

Figure 8.10: Deriving age changes for one distance measurement by mapping the actual measurement into the probability distribution. Table: [Far94]

vidual characteristics of the source head model even over large changes in age. Figure 8.10 demonstrates the mapping from one age to another.

For deforming the head geometry, the task at hand is: given the current landmark positions $\mathbf{p}_i$ $(i = 1, \ldots, N)$ and a number of distance measurements, what are the new landmark positions $\mathbf{q}_i$? This problem is largely underdetermined, as there are many solutions that fulfill the *hard constraints* imposed by the new distance measurements for the target age. To alleviate this, more constraints are added between landmarks that are close to each other: the distances between them should scale roughly with the global scaling s of the head, which is derived 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.

Linear Constraint Resolution

Most of the distance measurements are given along one axis a $(a \in \{x, y, z\})$, which allows to represent the problem as a set of linear constraints: find the n new landmark coordinates $\mathbf{q}_{a,i}$, $i = 1, \ldots, N$, for each axis a. The problem is solved separately for each axis. The relation to the M hard distance constraints and $\widetilde{M}$ soft distance constraints can be expressed by a sparse linear system:

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

where the combinatorial matrix $\mathbf{A} \in \mathbb{R}^{(M+\widetilde{M})\times N}$ 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 difference $\mathbf{q}_{a,j} - \mathbf{q}_{a,k}$. There can be at most $N(N-1)/2$ pairings of landmarks, but in practice only $M \approx 35$ plus $\widetilde{M} \approx 100$ neighboring landmarks are paired per axis. The vector $\mathbf{d} \in \mathbb{R}^{M+\widetilde{M}}$ represents the hard constraints and has M non-zero entries in those positions where the target distance is prescribed. Each vector $\widetilde{\mathbf{d}}_i \in \mathbb{R}^{M+\widetilde{M}}$ contains a single non-zero entry with the current distance

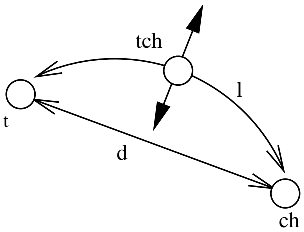


Figure 8.11: Arc-length measurements are used to modify face shape after solving constraints. The landmarks t and ch are constrained by their Euclidean distance d. The middle landmark tch is placed in the middle of a circle segment that approximates the desired arc length l.

$||\mathbf{p}_{a,j} - \mathbf{p}_{a,k}||$ 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 $\mathbf{q}_{a,j}$ and $\mathbf{q}_{a,k}$. Since I want to enforce the hard constraints $\mathbf{d}$ given by the data, but keep the relations between the other landmarks as unchanged as possible, the system is solved for weights λ_i close to the global scaling factor s. To this end, the equation is reformulated by shifting terms to include the $\widetilde{\mathbf{d}}_i$ and λ_i in the matrix:

$$\begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{D}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{q}_a \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{s} \end{pmatrix}, \tag{8.2}$$

where the columns of $\widetilde{\mathbf{D}} \in \mathbb{R}^{(M+\widetilde{M})\times\widetilde{M}}$ are composed of the vectors $-\widetilde{\mathbf{d}}_i$, and λ is a vector built from the λ_i in the same order. The submatrix $\mathbf{I} \in \mathbb{R}^{\widetilde{M}\times\widetilde{M}}$ is an identity matrix. On the right hand side, $\mathbf{s}$ has $\widetilde{M}$ entries with the constant scaling factor s. The system is now overconstrained and solved for the $\mathbf{q}_{a,i}$ and λ_i using a singular value decomposition (SVD) [PTVF92]. The singular values are clamped to achieve a prescribed condition number for the least squares problem before back-substitution. This removes linear dependent constraints, as they occur in practice.

According to this method, I 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. SVD will give a best fit in the least squares sense, though: 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 Equation (8.2), 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, I found the hard constraints to be fulfilled with a maximum absolute error of about two millimeters in the experiments.

Non-linear Constraints

For some landmark pairs a Euclidean distance is given, which can not be included directly into the equation system. I linearize this 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, it is assumed that the direction of that vector will not change drastically in the solution. This vector is scaled to the target length and projected onto the three axes of the global coordinate system. The three projected distances are added as additional linear constraints into the equation systems described in the previous section. Letting SVD run as before, a solution is found that approximately fulfills the Euclidean distance constraint. To improve this solution, the projection process is repeated, solving again until convergence. In practice, three iterations suffice.

Arc length is another measurement type in the data. Since the pairs connected by arcs are constrained additionally by distance measurements in the sample data set, it is not necessary to include the arc lengths into the constraint resolution mechanism. Instead, the arc measurements are used to improve the shape of the surface *after* solving. Approximating the arc by a circle segment, an additional landmark is placed between the two landmarks connected by the arc, as shown in Figure 8.11. This landmark is shifted along the surface normal to give the arc the desired length for the target age.

8.5.5 Discussion

The computed deformation of the face for changing its age produces plausible results, which is encouraging given that only a small amount of statistical data is used. Age progressions of two male individuals are depicted in Figures 8.12 and 8.13. The algorithm deals only with geometric deformation, textures are not automatically adapted. The figures thus show the deformed heads without textures and also with manually adapted textures.

My approach is inspired by DECARLO *et al.*[DMS98], but has no restrictions on the smoothness or parameterization of the surface. The computation of the age deformation uses only the landmark set, thus being independent from the type of the deformed surface—here, the deformation is applied to the vertices of the head model's triangle mesh.

The algorithm is purely statistical in nature, thus accurate predictions about individual head shape changes are difficult. It is, for instance, assumed 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. It would be useful to have data about the shifting of measured distances within the probability distribution over time.

Probably due to the error-prone physical measurement process carried out by many examiners over the years, the collected tabulated measurement data is not flawless. For instance, examination of the tabulated arc measurements delivered a surprising result: 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 children's cheeks that can be observed in the real world, I allowed for an adjustable slight increase in the arc length of up to 10 % over the original values for a young child. With today's modern scanning technology, it would be feasible to obtain much more detailed and precise measurements of human heads,

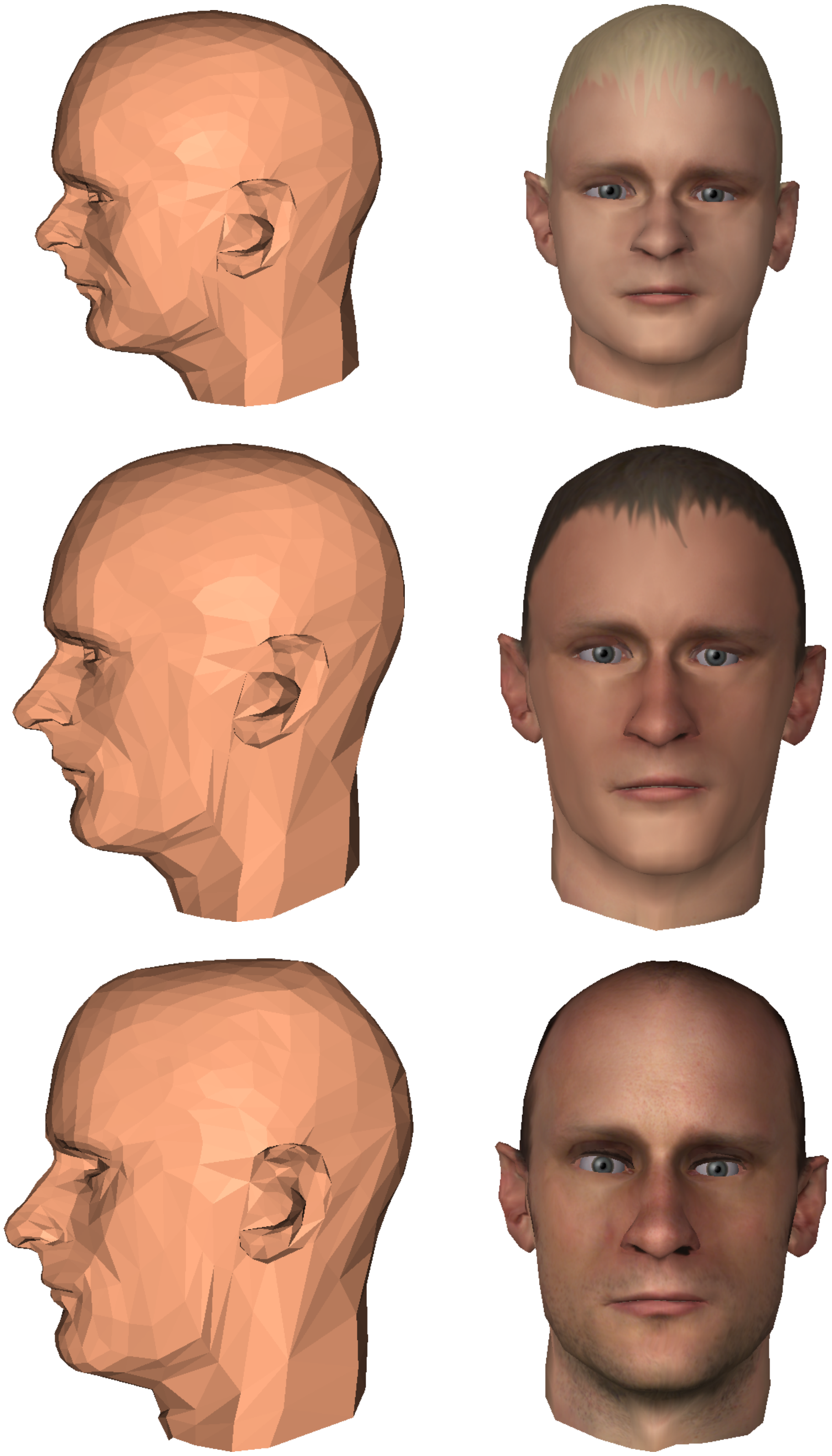


Figure 8.12: Deformation of adult head by the constraint resolution technique. Top to bottom: 5 years, 15 years, 31 years (original age). The textures on the models in the right column have been created manually from the original age photo texture.

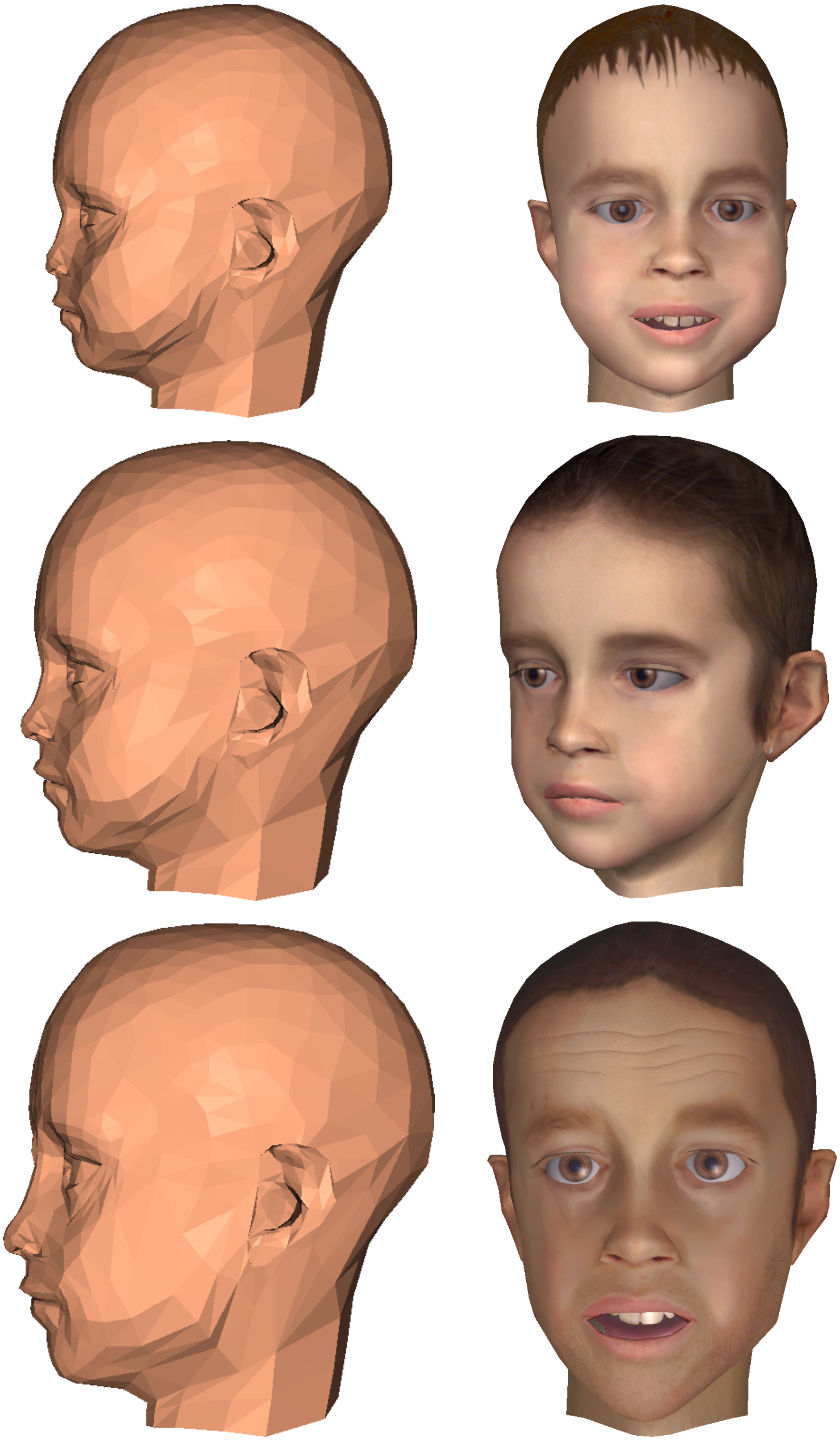


Figure 8.13: Deformation of a boy's head by the constraint resolution technique. Top to bottom: 1 year, 5 years (original age), 20 years. The right columns shows animated and textured versions of these models.

avoiding such workarounds and improving the results of the presented age-mapping algorithm.

There are many more aspects to growth and aging of faces. Wrinkle formation and changes in skin color or hair growth are not covered in the procedure presented here. The mechanical skin properties can be adjusted to a limited degree, though: the skin stiffness constants, skin thickness, and muscle layer thickness are evaluated during construction of each head model to initialize the spring mesh, the skull, and the muscle layer. These parameters can be adjusted manually to accommodate for individual and age-related changes. While it should be possible to have these values computed automatically, a thorough examination of the visual impact of such adjustments onto the actual animation is still required.

The contribution of the approach presented here is the simulation of the shape changes of the human head during growth. In combination with skin aging techniques such as those listed in Section 8.5.1, the method can be a substantial part in the development of a complete facial aging system.

8.6 Application 3: Face Reconstruction from the Skull

For well over a hundred years, forensic art and science has been assisting law enforcement. One of the major areas of concern in this area is facial reconstruction for postmortem identification of humans from their physical remains. Manual reconstruction and identification techniques build on the tight shape relationships between the human skull and skin: for instance, the presumed identity of a murder victim can be confirmed by superimposing a facial photograph with a properly aligned and sized image of the skull. If no photograph is available, the look of the face can be reconstructed to a certain degree by modeling the missing tissue layers directly onto the skull or a plaster cast made from it.

Figure 8.14: Historic reconstruction of a face from the skull. Left: sculpted reconstruction; right: corpse from which the skull was obtained. Images: [Tay01].

Figure 8.15: Comparison of sculpted reconstructions with photographs. Left: male subject; right: female subject. Images: [HRPM93].

The first documented case using three-dimensional facial reconstruction from the skull dates back to 1935 [Tay01]. A key experiment was later performed by KROGMAN [Kro46]: given the body of a deceased person, he took a picture of the cadaver head before extracting the skull. The skull was provided to a sculptor along with information about sex, origin, and age of the late owner, plus data on the average tissue thicknesses at several positions in the face. From this material, a reconstruction sculpture was created that could be compared to the original head. The result is shown in Figure 8.14. Since that time, three-dimensional facial reconstruction from the skull has been much refined, but the method has essentially remained the same. Researchers have examined the skull / skin relationships for different ethnic groups [LBV93] and analyzed the correspondences of skull morphology and facial features [FN93]. Others found correlations between muscle activity and skull shape [ML74, WH86]. In her comprehensive textbook, TAYLOR [Tay01] describes the craft in great detail.

Much of the fascination of the topic is due to the combined efforts of science and art, resulting in often astonishingly lifelike reconstructions, given the little available input (see Figure 8.15). Many parameters of the outward appearance of an individual cannot be readily derived from the skull, though. The process is thus highly dependent on rules of thumb, the experience of the artist, and some guesswork. It is, for instance, next to impossible to reconstruct the shape of the ears based on scientific reasoning, although empirically there seems to be a relation of ear height to the length of the nose [Tay01, p. 403].

8.6.1 The Manual Reconstruction Process

The traditional work process for facial reconstruction begins with preparation of the skull. Since the skull is often evidence in a criminal case, great care needs to be taken in handling it: some parts are extremely thin and fragile, especially in the nose and the orbits. For identification, the teeth often provide a lot of useful information, so a dental analysis is usually performed at this stage. For the reconstruction of the lower face, the mandible needs to be properly aligned and secured to the skull. It is important to take the natural spacing at the temporomandibular joint into account when doing this, which can be achieved using spacers. In cooperation with an anthropologist, and

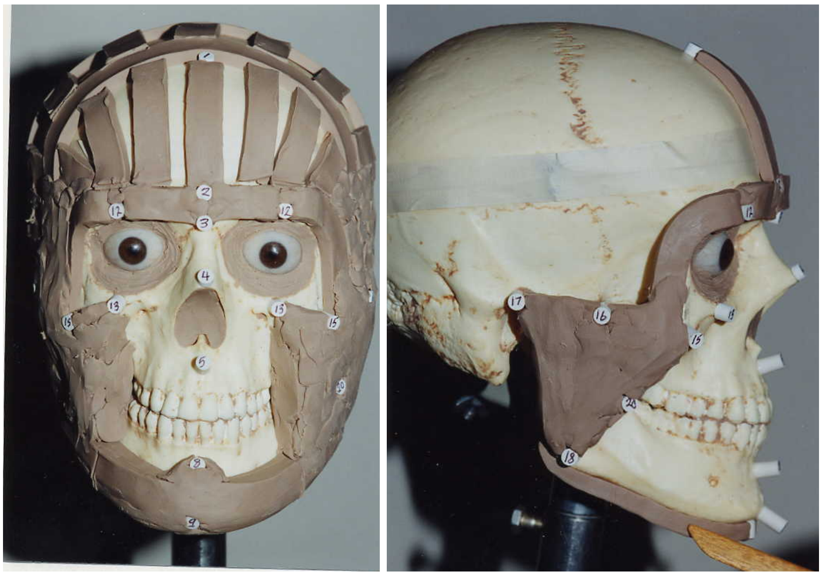


Figure 8.16: Modeling the face with clay on top of the skull using the tissue depth method. Images: [Tay01].

possibly given more information from the remains of the victim, an estimation of age, ancestry, sex, and stature can now be obtained.

The actual face reconstruction proceeds with one of two available approaches: the *anatomical method* and the *tissue depth method*. The anatomical method attempts reconstruction by sculpting muscles, glands, and cartilage, fleshing out the skull layer by layer. This technique is more often used in the reconstruction of fossil faces, where no statistical population data exists [ZPM98]. As TAYLOR states, this technique is very time consuming, occupying "many hundreds of hours". It also requires a great deal of detailed anatomical knowledge. Therefore, the alternative tissue depth method has become the more popular reconstruction technique in law enforcement. Here, standard sets of statistical tissue thickness measurements at specific points on the face are used. Each measurement describes the total distance from the skin surface to the skull, including fat and muscle layers. The method is thus more rapid than the anatomical method and does not require as much anatomical knowledge. Such measurements have been collected for males and females of several racial groups, using needles, X-rays, or ultrasound techniques. The tissue depth data most often used by police artists today was collected primarily by RHINE *et al.* [RC80, RM84]. The data is sorted into "slender", "normal", and "obese" groups, as well as by sex and race.

Given the set of measurements, tissue depth markers are now placed on the skull or a cast made from it, reflecting the tissue thickness at the sample points. These markers are oriented orthogonally to the skull surface, corresponding to the direction of the tissue thickness measurements. Using the markers and other features on the skull for guidance, the face is modeled on top of the skull using clay. A snapshot of the beginning stages of a reconstruction using the tissue depth method is shown in Figure 8.16.

8.6.2 Developing a Computer-Based Approach

The tissue-depth method as described above translates remarkably well to my head modeling framework: in the applications presented in Sections 8.4 and 8.5, the generic face mesh with underlying muscle and bone layers is deformed to match given target constraints on the skin. This process is adopted here to match the muscle and skin layers to given skull data instead. The manual "dowel placement" method is implemented by extending the skull landmark editor briefly discussed in Section 5.1.1. The sculpting of the skin surface is performed as shown before by applying the RBF-based volume deformation to a head model template. The deformation approach has the additional advantage of being applicable to the additional structures attached to the template: the muscle structure is mapped to the fitted head model as well, enabling animation on the reconstructed head in the physics-based facial animation system [KHS03].

After reviewing related work in the following section, acquisition of skull data and interactive landmark placement are discussed in Section 8.6.4. Section 8.6.5 describes how the generic head model is fitted to the skull. Animation and texture generation for the resulting head model are touched upon in Section 8.6.6. I present some examples in Section 8.6.7 and draw conclusions from these results in Section 8.6.8.

8.6.3 Related Work

Perhaps due to the lack of rigid taxonomies and hard rules, the use of computers and computer graphics in this forensic application is still very limited. The procedures described above cannot be cast easily into a computer program that produces good results in an automated manner—the experience and judgment of the practitioner remain a vital part of the system.

In law enforcement practice, computer-aided techniques restrict to relatively simple image and video manipulation: face photographs are used for skull superimposition [Grü93, MYIS95], while image warping and retouching enable a basic simulation of aging [Tay01, p. 253]. This situation is unfortunate, since the traditional three-dimensional face reconstruction process is extremely time-consuming and expensive. It is hardly feasible to produce a variety of different plausible reconstructions from one skull, simply due to the effort that has to be put into the creation of each model. Also, repeated physical handling of the original skull increases the risk of damage.

One prototypical computer-based face reconstruction system, allowing fitting of a generic hierarchical B-spline head model to a skull mesh, is described by ARCHER in her Master's thesis [Arc97]. The user places dowels on a skull model with prescribed tissue thickness values, resulting in targets for a B-spline surface fitting process. The interpolation process is tricky and requires careful preparation of the template head model.

In the approach presented by MICHAEL and CHEN [MC96], a source head model that includes a skull is deformed using a volume distortion function such that the deformed source skull approximately matches the target skull. It is assumed that the deformed source head model bears good resemblance to the (unknown) target head model. The volume distortion function is set up as a field warp using forty pairs of disc fields, which are manually placed around the skull. No details are given on the placement of these control fields.

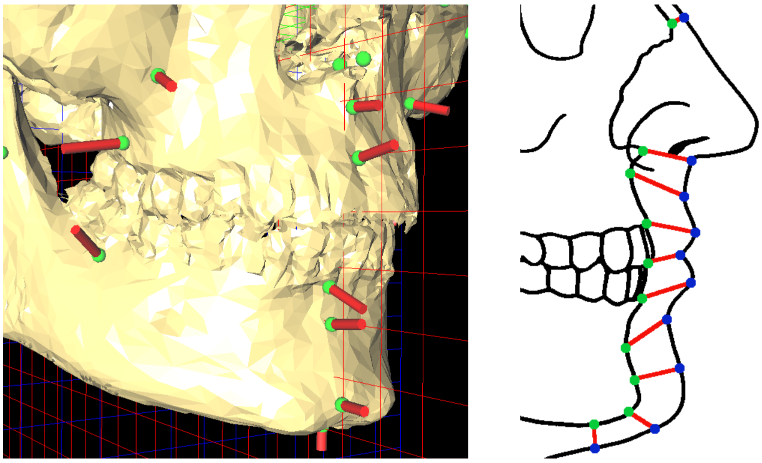


Figure 8.17: Skull landmark specification in the mouth area. Left: snapshot from landmark editor; right: correspondences between skull and skin markers, showing orthogonal and non-orthogonal skull/skin relationships. Right image after [yI93].

A deformation technique similar to the one used here is employed by VANEZIS *et al.* [VVMN00]. A facial template chosen from a database of scanned faces is deformed to match the position of target face landmarks, which have been derived from adding statistical tissue thickness values to the corresponding skull landmarks. The resulting reconstructed heads are not always complete (for instance, the top of the head is usually missing). The authors suggest to export an image of the reconstructed head and to apply a final image-processing step to add eyes, facial and head hair.

The above methods require a lot of manual assistance in setting up the interpolation function [Arc97, MC96], or rely on a database of head templates [VVMN00]. In contrast, with the approach presented here, reconstructions are developed from one head template with relatively few markers, and additional mechanisms are used to improve reconstruction results (see Section 8.6.5). The approach always generates complete head models. Instead of using higher-order surfaces or point samples, the surface of the deformable head template is an arbitrary triangle mesh, simplifying later artistic modifications of the result using standard modeling tools. Integration of expressive facial animation has not been discussed by other computer-aided facial reconstruction approaches so far.

8.6.4 Preparation of the Skull

My approach uses three-dimensional skull data acquired, for instance, from volume scans and extraction of the bone layers, or by range scanning a physical skull. The test data used for the examples in Section 8.6.7 was acquired using both types of scans. To speed up further processing, a low-resolution triangle mesh comprised of 50-250k polygons is produced from the skull model by mesh decimation techniques [GH97]. In general, the original data should be simplified as little as possible since minute details on the skull can give important clues for the reconstruction. The mesh resolution is chosen for adequate responsiveness of the interactive skull editor. In practice, it is helpful to have the original data set (or the physical skull) ready as a reference during

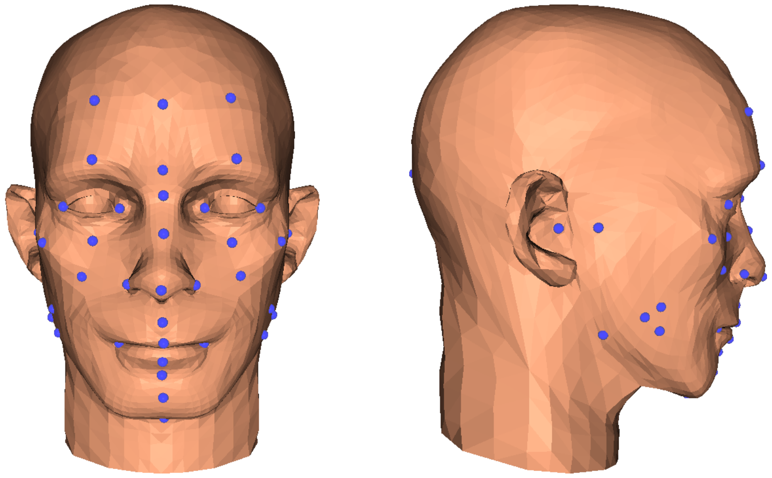


Figure 8.18: Variation of the deformable head model used for reconstruction. The skin landmarks paired to skull landmarks are shown as blue dots. Left: front view; right: side view.

editing.

In the editor, the skull model is equipped with landmarks, as shown in Figure 8.17. Points on the skull surface are simply picked to create a landmark, which can then be moved around on the surface for fine positioning. Each landmark is associated with a vector in surface normal direction, corresponding to the typical direction of thickness measurements. As can be seen on the right image in Figure 8.17, some skull / skin correspondences are in fact non-orthogonal to the skull surface in the area of the lips. This is corrected for at a later step of the fitting process, as described in Section 8.6.5. The landmark vector is scaled to the local tissue thickness, which is looked up automatically by the landmark's assigned name in a table based on RHINE's data (see Section 8.6.1). The specific set of paired landmarks on skin and skull used in this application is listed in Table 8.3. Most skull landmark names and descriptions are taken from [Tay01, p. 350 ff.]. Short skull landmark names are listed in the *id* column. I have tried to adhere to naming conventions used in the forensic and anthropometric literature as much as possible [Tay01, yI93, Far94]. For simplicity, I have assigned the same short name to corresponding landmarks on skull and skin, which is not generally the case in the literature. In a few cases, marked by $*$ in the table, I invented short names. Not all skull landmarks have an "official" counterpart on the skin, so I placed the corresponding skin markers using my own judgment. The *mp* landmark pair is not part of the standard set. It was added to improve the alignment of skin to skull in the region behind the ears, where the mastoid process adds a bulge to the skull.

8.6.5 Fitting the Deformable Head Model

Head Model Structure

When the skull is tagged with landmarks, it serves as the target for deformation of the generic head model shown in Figure 8.18. This model is a slight variation of the one described in Section 5.1. Most notably, a separately controllable jaw as well as the eye,

name	*id*	*description*
		Midline
Supraglabella	tr	Above glabella, identified with the hairline
Glabella	g	The most prominent point between the supraorbital ridges in the midsagittal plane
Nasion	n	The midpoint of the suture between the frontal and the two nasal bones
End of nasals	na	The anterior tip or the farthest point out on the nasal bones
Mid-philtrum	a	The mid line of the maxilla (east and west), placed as high as possible before the curvature of the anterior nasal spine begins
Upper lip margin (Supradentale)	sd	Centered between the maxillary (upper) central incisors at the level of the Cementum Enamel Junction (CEJ)
Lower lip margin (Infradentale)	id	Centered between the mandibula (lower) central incisors at the level of the Cementum Enamel Junction (CEJ)
Chin-lip fold (Supramentale)	b	The deepest mid line point of indentation on the mandible between the teeth and the chin protrusion
Mental eminence (Pogonion)	pog	The most anterior or projecting point in the mid line on the chin
Beneath chin (Menton)	me	The lowest point on the mandible
		Bilateral
Frontal eminence	fe*	Place on the projections at both sides of the forehead
Supraorbital	sci	Above the orbit, centered on the upper most margin or border
Suborbital	or	Below the orbit, centered on the lower most margin or border
Endocanthion	en	point at the inner commissure of the eye fissure; the landmark on the skin is slightly lateral to the one on the bone
Exocanthion	ex	point at the outer commissure of the eye fissure; the landmark on the skin is slightly medial to the one on the bone
Inferior malar	im	The lower portion of the maxilla, still on the cheekbone
Lateral orbit	lo	Drop a line from the outer margin of the orbit and place the marker about 10 mm below the orbit
Zygomatic arch, midway	zy	Halfway along the zygomatic arch (generally the most projecting point on the arch when viewed from above)
Supraglenoid	sg	above and slightly forward of the external auditory meatus
Gonion	go	The most lateral point on the mandibular angle
Supra M^2	spm2*	Above the second maxillary molar
Occlusal line	ol	On the mandible in alignment with the line where the teeth occlude or bite
Sub M_2	sbm2*	Below the second mandibular molar
Mastoid process	mp*	Most lateral part on the mastoid process behind and below the ear canal

Table 8.3: Landmark set used for face reconstruction

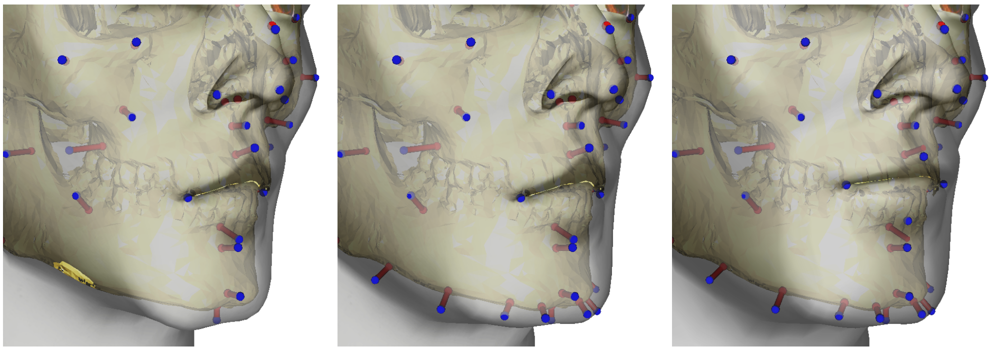

(a) (b) (c)

Figure 8.19: Fitting stages, shown on the lower face. a) using only user-specified landmarks (some skull areas still intersecting the skin); b) with automatically interpolated landmarks on the mandible; c) using additional heuristics for lip and nose shaping.

teeth, and tongue models have been left out of the reconstruction approach since the added animation capabilities provided by these features are not particularly useful for this application: while a modest change of expression such as a smile or a frown might aid identification, rolling of eyes, blinking, and talking would probably not. It is also nearly impossible to correctly guess details such as a specific way of speaking—errors in this respect would produce rather misleading results in a real identification case. In conclusion, the effort of providing a separate jaw mesh, as well as placing tongue, eye, and potentially teeth models does not offset the benefits. Likewise, real-time animation and rendering speed are not important for the reconstruction application. The head model has thus been refined to 8164 triangles to allow for a better fit of the geometry and higher quality rendering. The adapted head model and the landmarks defined on the skin surface are shown in Figure 8.18. The majority of the landmarks corresponds to the user-specified skull landmarks. These landmark pairs control the basic fitting of the head structure, while a few additional landmarks only defined on the skin are used for the final adjustments of the reconstructed shape. This process is discussed in the following sections.

Deformation

Given the deformable head model with N predefined skin landmark positions $\mathbf{p}_i \in \mathbb{R}^3$ and the corresponding landmarks $\mathbf{s}_i \in \mathbb{R}^3$ $(i = 1, \ldots, N)$ specified on the skull, a space deformation is set up that fits the skin and the muscle layout to the skull.

The target skull landmarks have associated tissue depth vectors $\mathbf{d}_i$, so corresponding skin landmark positions $\mathbf{q}_i$ are defined as $\mathbf{q}_i = \mathbf{s}_i + \mathbf{d}_i$. The problem can now be treated as described in Section 8.3, by setting up a function $\mathbf{f}$ that maps the $\mathbf{p}_i$ to the $\mathbf{q}_i$. In contrast to the previous applications, the skull geometry is already given, so only skin mesh and muscles are transformed to match the template head model to the skull.

Since the landmark set is comprised of only 40 landmarks (see Table 8.3) and is thus very sparse, the computed deformation doesn't properly align the skin to the skull in all places, as can be seen in Figure 8.19(a). Interactive specification of more

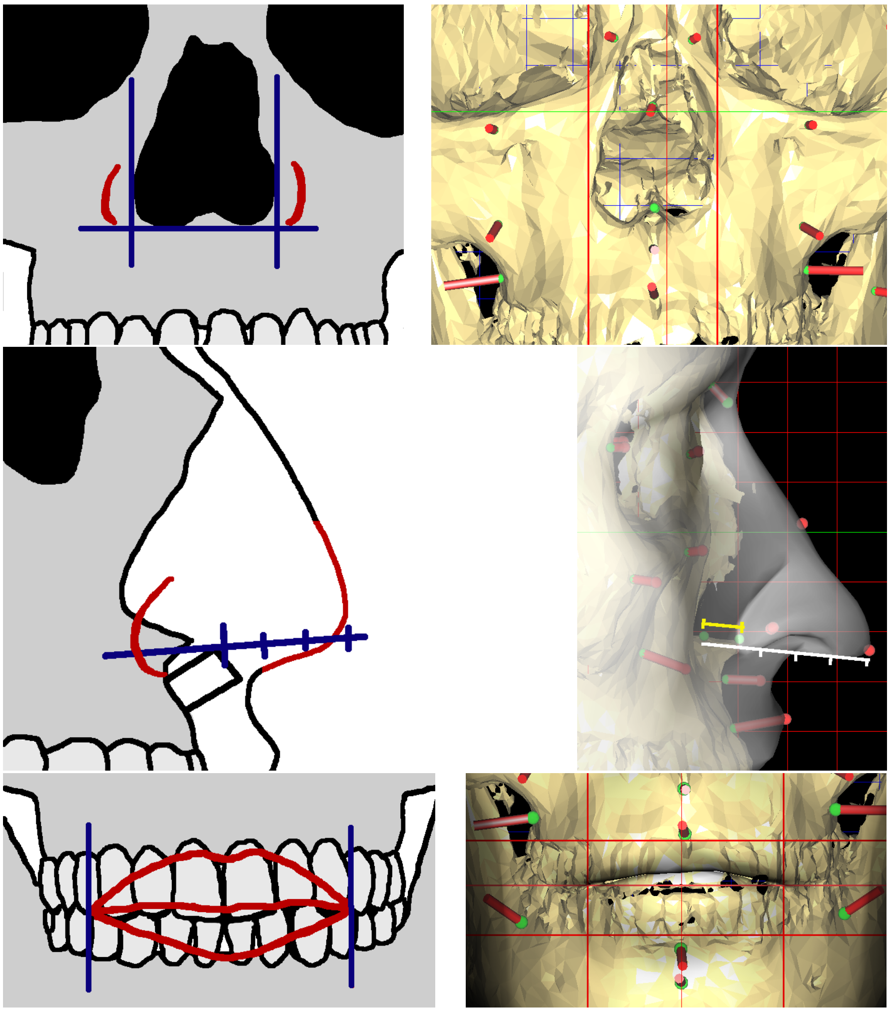


Figure 8.20: Comparison of heuristics used in traditional reconstruction (left) with the graphical interface (right). (Note: different skulls are used in the adjoining images.) Top: estimation of nose width; center: positioning of the nose tip; bottom: setting lip width, height, and mouth corner position.

landmarks puts an undesirable additional burden onto the user, so additional landmark pairs are computed automatically by interpolation between existing ones on the upper and back part of the cranium, as well as on the mandible, as shown in Figure 8.19(b). The thickness value of an interpolated skull landmark is also interpolated, where only such skull areas are chosen for landmark interpolation where the tissue thickness is near-constant. Tissue depth interpolation would be problematic, for instance, in the mid-face area, where thickness values change drastically from the cheekbone to the mid-face region below.

Additional Reconstruction Hints

The tissue depth values at the marker positions define the basic shape of the reconstructed head, assuming depth measurements being always strictly orthogonal to the skull surface. As mentioned in Section 8.6.4, this assumption is not always valid. A number of rules are thus used in traditional facial reconstruction to help locate certain features of the face based on the skull shape, employing empirical knowledge about shape relations between skin and skull [Tay01]. I have translated some of these heuristics for use with the skull landmark editor: the final fitting result, as shown in Fig. 8.19(c), is obtained by including this additional user input.

To keep the user interface uniform, most rules are expressed by the placement of vertical and horizontal guides in a frontal view of the skull. From this user input, the placement of a few landmarks on the skin is adjusted, resulting in a new target landmark configuration. The updated landmark set is used to compute another warp function, which deforms the pre-fitted head model in the adjusted regions. Five rules influence the shape of the nose and the shape of the mouth, as shown in Figure 8.20:

- The width of the nose wings corresponds to the width of the nasal aperture at its widest point, plus 5mm on either side in Caucasoids. In the editor, the user places two vertical guides to the left and right of the nasal aperture. From their position, the displacement of the two *al* skin landmarks placed at the nose wings is computed (cf. Figure 8.20, top row).

- The position of the nose tip depends on the shape of the anterior nasal spine. According to KROGMAN's formula [Tay01, p. 443], the tip of the nose is in the extension of the nasal spine. Starting from the z value of the tissue depth marker directly below the nose (mid-philtrum, see Table 8.3), the line is extended by three times the length of the nasal spine (cf. the white and yellow lines in the right image of Figure 8.20, middle row). In the editor, begin and end points of the nasal spine are marked. The *prn* landmark at the nose tip is then displaced according to the formula.

- The width of the mouth is determined by measuring the front six teeth, placing the mouth angles horizontally at the junction between the canine and the first premolar in a frontal view. Two vertical guides are used for positioning the *ch* landmarks located at the mouth angles (vertical lines in Figure 8.20, bottom row).

- The thickness of the lips is determined by examining the upper and lower frontal teeth. Seen from the front, the transition between the lip and facial skin is placed at the transition between the enamel and the root part of the teeth. Two horizontal guides are placed by the user at the upper and lower transition, respectively. This determines the vertical position of the *id* and *sd* landmarks marking the lip boundary (top and bottom horizontal lines in Figure 8.20, bottom row).

- The parting line between the lips is slightly above the blades of the incisors. This determines the vertical placement of the ch landmarks (middle horizontal line in Figure 8.20, bottom row).

Using these heuristics, a better estimate of the mouth and nose shapes can be computed. The effect is strongest on the lip margins, since the assumption of an orthogonal connection between corresponding skin and skull landmarks is in fact not correct at these sites, as the right part of Figure 8.17 shows. The initial deformation thus gives a

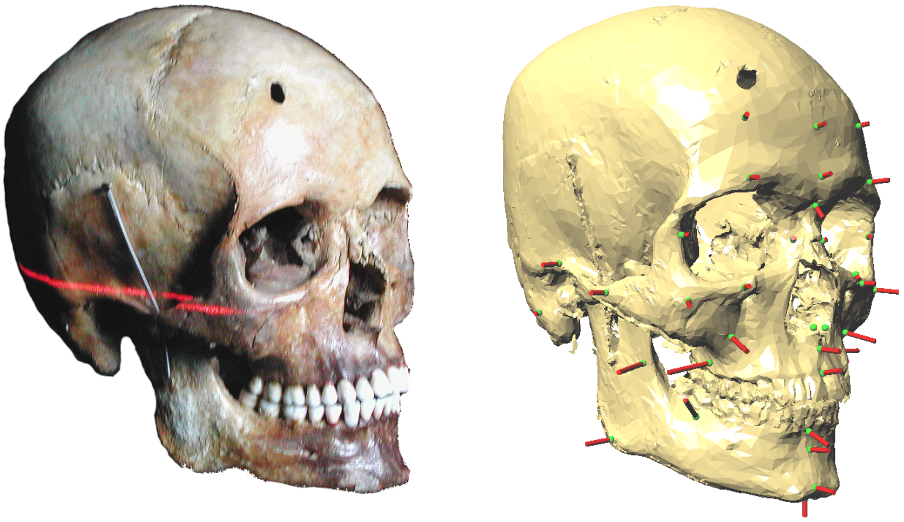


Figure 8.21: Preparation of a real skull for reconstruction. Left: the skull is laser-scanned; right: resulting skull mesh, simplified and tagged with landmarks.

good estimate of the tissue thickness of the lips while the second deformation using the information provided by interactive guide adjustment refines the vertical placement of the lip margins.

8.6.6 Facial Expressions and Rendering

In manual facial reconstruction, a neutral pose of the face is preferred as the most "generic" facial expression. Other expressions could be helpful for identification purposes, but the cost of modeling separate versions of the head model is prohibitive. In the virtual reconstruction approach presented here, this does not pose a problem: since the fitted head model has the animatable structure of skin and muscles, different facial expressions can be assumed by setting muscle contractions.

If additional information about the modeled person is available, for instance, from remnants of hair found with the skull, the resulting mesh can be colored correspondingly. The system includes basic capabilities for coloring the parts associated with skin, lip, and eyebrows in the model's texture map. Colors can be adjusted interactively in HSV space on the reconstructed head model. Finally, the color adjustments are merged into a neutral base texture and saved as a new texture map. The fitted, texture-mapped triangle mesh can be easily imported into various rendering packages for display. The examples shown in Figures 8.22 and 8.23 show three different skin colorations created in this way.

8.6.7 Reconstruction Examples

I have tested the technique on a real skull and on two medical volume scans, made available by the Institute of Forensic Medicine, Saarland University. All data pertains to individuals of Caucasian type. Each reconstruction required approximately an hour of interactive work, excluding time for data acquisition.

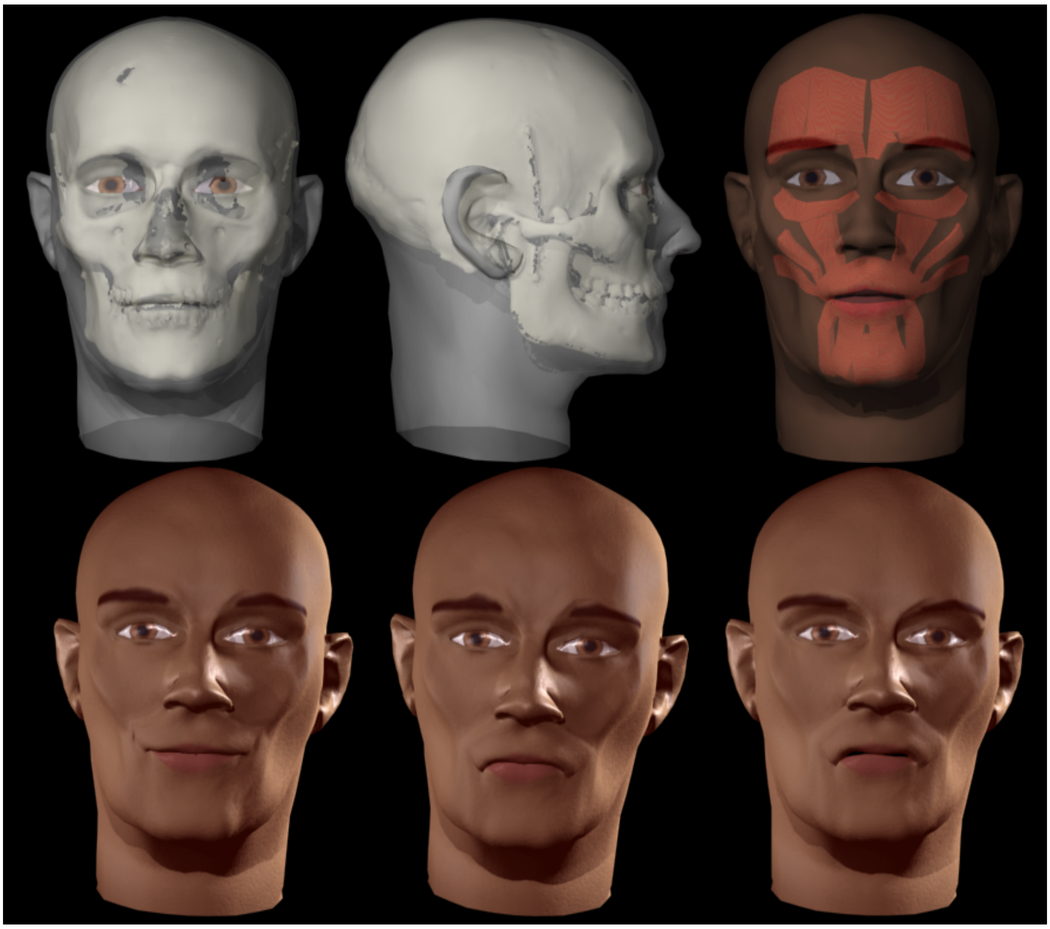


Figure 8.22: Example of facial reconstruction created from a scanned real skull, showing fit of skin to skull, transferred muscles, and three facial expressions.

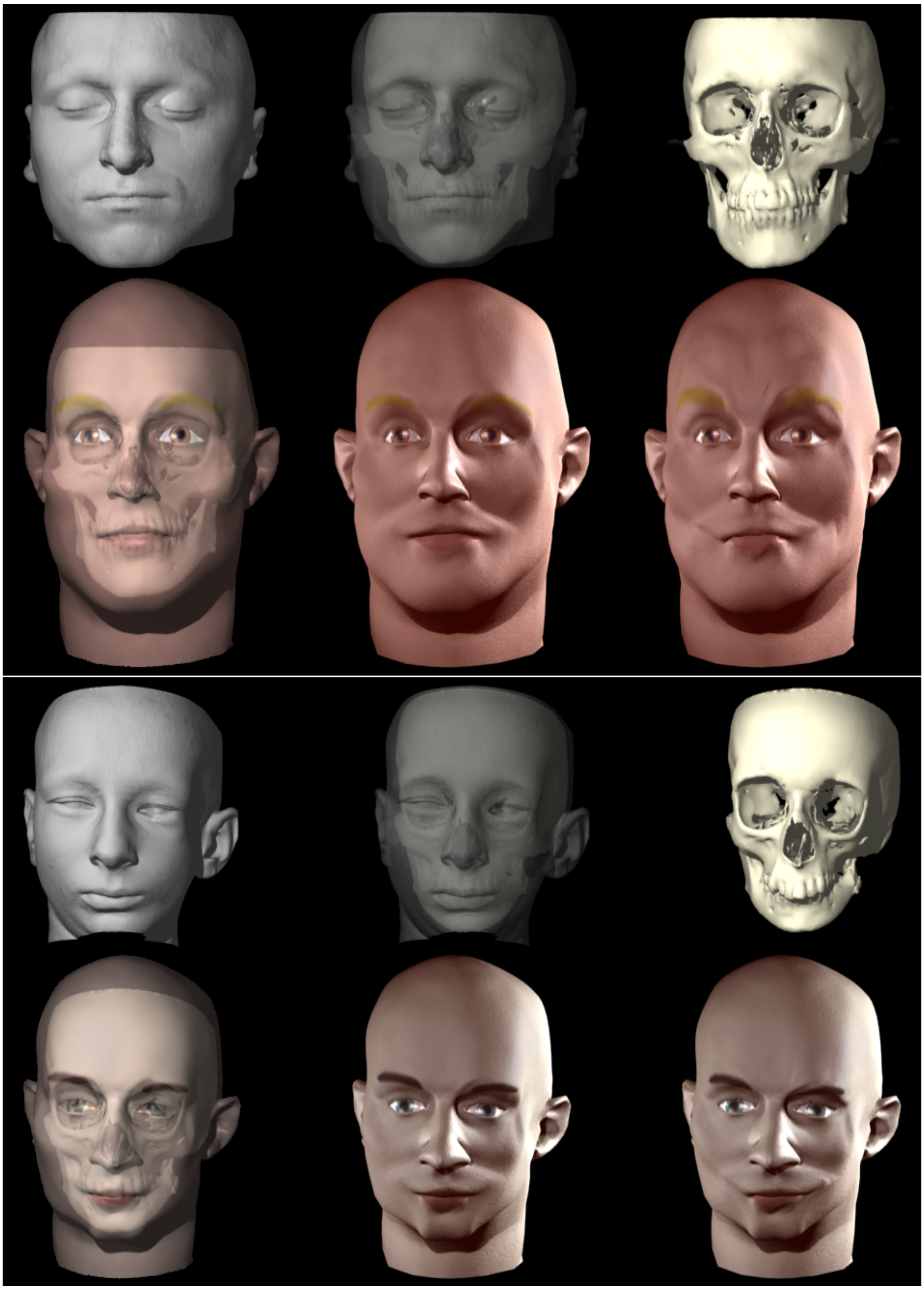


Figure 8.23: Examples of facial reconstructions from volume scans. Top: male subject, showing the actual face as contained in the data, superimpositions of the actual and the reconstructed face with the skull, and the reconstruction with neutral and "worried" expression. Bottom: female subject with strong skull deformations. The CT data sets don't contain the top and bottom of the heads, thus the source skull and face models are cut off. The actual head height had to be guessed in these cases.

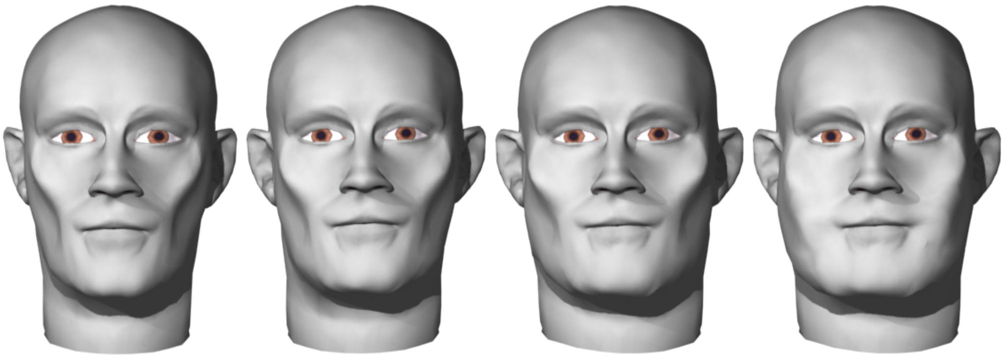


Figure 8.24: Left to right: RHINE's traditional "slender", "average", and "obese" tissue depth tables (cf. [Tay01, p. 350 ff.]) often result in hollow cheeks and prominent cheekbones (see also Fig. 8.22 and 8.23). Rightmost image: the shape can be improved by "bulging out" the affected mesh areas.

The real skull, shown in Figure 8.21, was unearthed on a construction site and belongs to an unidentified male, approximately 35 years of age. As can be seen from the hole in the frontal bone, he was killed by a head shot. It is believed that the owner of this skull was a war victim or a soldier.

After scanning the skull, the resulting mesh was simplified to 100k triangles. The interactive placement of the skull landmarks and facial feature guides was relatively easy in this case since the skull is complete and in good condition. The skull mesh with user-specified landmarks is depicted in Figure 8.21. Due to its war-time origin, the face can be assumed to be rather skinny, so I selected the "slender" tissue thickness table. Fitting results can be seen in the top half of Figure 8.22. Since the actual appearance of the individual is unknown, the accuracy of the reconstruction can only be guessed. Nonetheless, the reconstruction seems plausible. Notably, the shape of the chin, which can be predicted from the corresponding region on the skull, has been reproduced well.

To show examples utilizing other data sources, and also for validation, skull and skin surfaces were extracted from medical volume scans. The first data set, shown in the top row of Figure 8.23, pertains to a male subject of roughly 30 years. The subject's face is rather bulky, so the "obese" tissue thickness data set was chosen (in a real case, this choice would have to be made based on other available information such as the size of clothes, if present). The first reconstruction attempts showed a consistent emphasis on prominent cheek bones and hollow cheeks: no matter which data set was picked, the face would become more bulky, but not show the expected general roundness of the face. This effect is demonstrated in Figure 8.24 on variations of the first model. A closer examination revealed that the reason lies in the relatively low thickness values RHINE assigned to the landmarks defining the cheek region (*sbm2* and *spm2* in Table 8.3). After excluding these two landmarks, I obtained the results shown in Figure 8.23. The rightmost image in Fig. 8.24 shows how simple mesh modeling techniques could be used at this point to improve and individualize the reconstruction.

The second volume data set shows a female patient with strong skull deformations. A reconstruction of this face was produced to test the method with a decidedly non-average skull shape. The result can be seen in the bottom half of Figure 8.23. Since the automatic landmark interpolation scheme (see Section 8.6.5) is designed to handle the normal range of skull variations, the unusual shape of the mandible resulted in very sparse sampling of the chin area. A prominent feature of the skull data is the protrusion of one incisor, pushing the upper lip to the front. To model this effect, I moved the *sd* landmark a few millimeters down onto the blade of the incisor, thus pushing the associated skin landmark forward as well. This did not have an adverse effect on the definition of the height of the upper lip boundary since this is adjusted separately by the mouth guides (cf. Figure 8.20).

8.6.8 Discussion

The face reconstruction approach presented in this section mirrors the manual tissue depth method and thus has essentially the same prediction power. The results show overall good reproduction of facial shape and proportions, and some surprisingly well-matched details. It should be noted that the examples were produced by a computer scientist with no training in forensic reconstruction.

The advantages of the computerized solution are evident: instead of weeks, it takes less than a day to create a reconstructed face model, including scanning of the skull. Once the scan data is marked with landmarks, different varieties such as slimmer or more obese versions can be produced within seconds at the push of a button, which is practically impossible with the manual method due to the vast amount of time needed for production of a single model. Slight variations in facial expression can also be obtained quite easily by animating the muscle structure underlying the model.

Since the virtual reconstruction is based on 3D scans, which can be acquired contact-free, the risk of damage to the original skull is reduced. On the other hand, the scanning process has inherent limitations: depending on the maximum resolution of the digital scanner, much of the finer detail on the skull is lost. The delicate structure of, for instance, the nasal spine cannot be fully captured with current scanning technology. For this reason, it is necessary to consult the original skull from time to time for reference.

In my experiments, I often found that surface normals on the scanned skull geometry do not always behave the way they should, reflecting the orientation of the surface only very locally. It might be useful to consider an average of normals in a larger area around the landmark position to solve this. Sometimes, it would be desirable to adjust the orientation manually.

The interactive system allows for an iterative reconstruction approach: a model is produced quickly from a given landmark configuration, so landmarks can be edited repeatedly until the desired result is obtained. The emphasis on the interaction component makes the speed of the fitting process an important issue. While the actual calculation of the warp function and the deformation of the mesh are performed instantaneously, about five seconds are needed in the test setting on a 1.7 GHz Pentium Xeon to examine skull and skin for potential insertion of additional landmarks. This time is for the largest part used for ray intersections of the skull and skin meshes, which are done in a brute force manner. A big speed-up can be expected through the use of space partitioning techniques.

For practical use, the facial reconstruction system should provide more editing facilities for skin details and hair. Useful additions include, for instance, a choice of templates for haircuts and facial features such as eyebrow shapes, beards, and wrinkles.

At this point, large-scale validation of the system would be necessary to evaluate the usability of the system.

As TAYLOR writes in her book, the tissue depth values should not be taken at face value in three-dimensional facial reconstruction, but rather act as guides for the final facial reconstruction, which still relies heavily on artistic skills and intuition. My preliminary tests confirm that strict adherence to RHINE's data for the solution of the interpolation problem is too limiting. This indicates not a weakness in the method, but reflects the low number of samples (between 3 and 37 in each group) and the technical limitations at the time RHINE assembled his data tables. Given the current state of technology, more samples of higher precision could be acquired, resulting in much more comprehensive and usable data. Ultimately, the computer-based facial reconstruction approach could then even become superior to the traditional approach.

8.7 Discussion

The applications presented in the previous sections show the potential of the head deformation framework. The internal muscle and bone structures of a derived head model are represented in an anatomically plausible manner, enabling techniques that use actual anthropometric measurements on skin and skull to produce a fully animatable model.

As a useful side effect of always using the same generic model, all derived heads share a common parameterization, i.e., the same set of muscles. Expressions and animations can thus be transferred between models, which has shown to work well in practice and is useful for building a common expression and animation library. NOH [NN01] has presented a technique for transfer of motion between different head model geometries. A drawback of that approach is the loss of the original animation parameters, i.e., the target model cannot be controlled by the parameters anymore, thus making modification of the transferred animation difficult. This does not pose a problem with my approach, simplifying, for instance, incorporation of individual characteristics of facial motion—such as a particular way of smiling—into canned animation. There is some automation potential for dealing with age-related animation characteristics, based on the muscle parameterization: for instance, for a young child, facial muscle movements can be globally scaled down to accommodate for the not yet fully developed abilities of expression.

The results from the applications presented in this chapter indicate that *anthropometric modeling* is a fruitful approach, and can be a useful tool for artists and scientists alike. Still, more detailed, reliable, and precise anthropometric data is needed. Based on the constraint resolution technique used for aging, presented in Section 8.5, new face modeling tools can be devised that allow specification of facial feature relations according to age, sex, ethnicity, body type, or other statistically captured variables. Apart from the applications shown, the basic deformation technique can also aid in other common tasks such as morphing between animated head models.

CHAPTER 9

Rendering

Show my head to the people, it is worth seeing.
– Georges Jacques Danton, to his executioner

The main contributions of this dissertation pertain to the modeling and animation of virtual head models. Still, an animation system is not complete without actually visualizing the animated geometry—in this case, in real time. The main drawback of real-time animation is the relatively coarse model resolution dictated by the numerical skin / tissue simulation: even using the dual-processor strategy described in Section 5.6, no more than about 2000 triangles can be used for the skin. While, at this level of complexity, the simulation thread keeps one CPU completely busy, there is still a lot of capacity left in the rendering thread on the other CPU. The facial components eyes, teeth, and tongue can thus be rendered with high resolution: they are simple geometric objects that do not add to the computational load in the simulation. It is desirable to likewise increase the resolution of the rendered skin geometry without adding complexity to the physics-based animation. This is achieved with several strategies in the animation system:

- hardware-accelerated shading;
- displaced subdivision surfaces on top of the animated face mesh;
- dynamic refinement of the geometry during animation.

The remainder of this chapter discusses these techniques.

9.1 Hardware-Accelerated Shading

The head model is in the simplest case rendered as a set of textured triangles using OpenGL, where the textures are generated from photographs (see Section 5.5). To improve the visual appearance of the skin surface without diminishing performance, the vertex program and register combiner features of the NVidia GeForce3 graphics board are employed, in particular, multitexturing using the four available texture units. These units are assigned to:

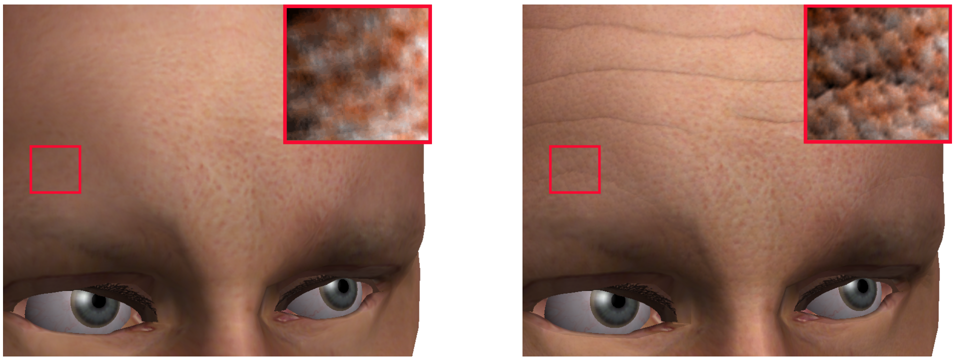


Figure 9.1: Snapshot of an animation, eyebrows raised. Left: plain OpenGL rendering; right: with additional real-time skin shading. The inset boxes are color-enhanced to visualize the different skin structure better in print.

1. the skin color texture;
2. a bump map for the skin surface structure, coded in the RGB color channels;
3. another bump map for expressive wrinkles, plus a mask containing intensity values for the surface structure bump map in the alpha channel;
4. a mask texture that contains three monochrome masks for specification of separate wrinkle regions via the color channels, plus a gloss map in the alpha channel.

The bump map for the skin structure is computed directly from a synthetic human skin model similar to the one presented by ISHII *et al.* [IYYT93]. The cell-like structure gives the skin a more defined look, as can be seen in the right image in Figure 9.1. For creation of the expressive wrinkle layout, the wrinkles are hand-drawn into a separate image, following the layout of the face in the skin texture. From this image, a bump map is automatically created. Hardware bump mapping for both skin structure and wrinkles is implemented using the OpenGL `NV_vertex_program` and `NV_register_combiners` extensions. The intensity of the skin structure bump map can be locally varied using an intensity map, which contains a scalar value per texel to specify the degree to which the bump mapped normal should affect the lighting computation. Considering the non-uniform specularity of facial skin, a gloss map is applied to specify local specular coefficients for the Blinn-Phong shading model.

The whole process is carried out in a single rendering pass on the GeForce3. The expressive wrinkles reside together in one bump map, divided into three distinct regions with separate intensity control. The intensity of the forehead part of the bump map is directly connected to the contraction value of the frontalis muscles, leading to convincing wrinkles on the forehead when the eyebrows are raised, demonstrated in Figure 9.1. Also, the intensity of the furrows between the eyebrows are linked to the corrugator supercilii muscles, which pull the eyebrows together. All other facial wrinkles have to reside together in the third available wrinkle region. Since their intensities cannot be controlled independently, they are not linked to facial muscles. Without resorting to multi-pass rendering, separate control over, for instance, the crow's feet at the eyes and the wrinkles of the mouth area will be possible when newer graphics boards with more texture units become available.

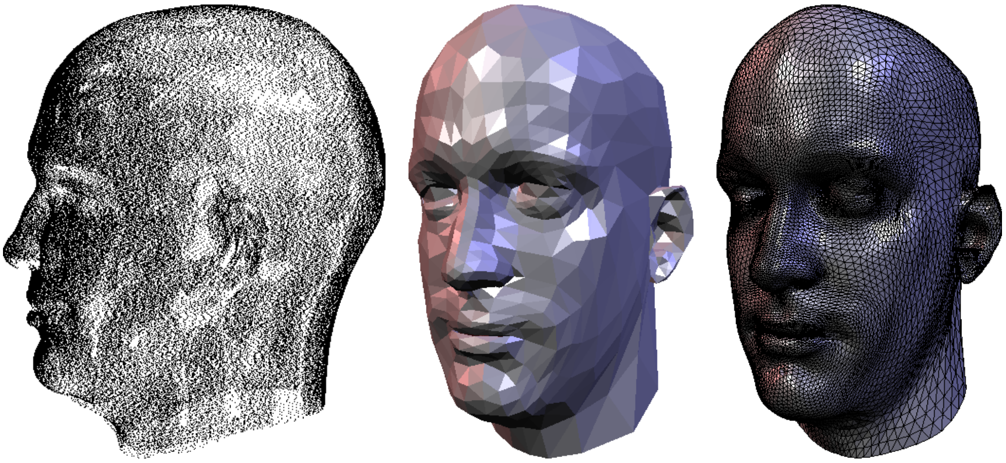


Figure 9.2: Displaced subdivision surface head model created from point cloud data. Left: Point cloud from scanner (thinned for this image); center: control mesh fitted to the points; right: level 2 subdivision surface.

9.2 Displaced Subdivision Surfaces

Even though proper shading of a polygonal surface can often deceive the eye successfully, it still falls short of using actual high resolution geometry. The limitations become obvious, for instance, at the silhouettes of an object. Also, the typical Gouraud shading model with Phong reflection at the vertices does not deliver crisp highlights on low resolution geometry. As noted above, the simulation part of the facial animation system does not allow an increase of the resolution of the *animated* mesh. It is well possible, though, to increase the resolution of a separate *rendered* mesh, constructed from the animation mesh and following its deformations. This can easily be done using subdivision surfaces.

Subdivision surfaces [ZS00] have become increasingly popular in modeling and animation due to their ability to bridge the gap between polygon meshes and higher-order surfaces. Since they are constructed by repeated refinement of triangle or quadrilateral meshes up to an arbitrarily close approximation to their smooth limit surface, they also provide an effective means to control accuracy and efficiency of rendering in a systematic manner. A *displaced subdivision surface* stores additional detail in the form of vertex offsets in surface normal direction [GVSS00, LMH00]. At every refinement level, the surface can thus reveal more and more surface detail. Given a high resolution input scan, such a surface can be generated automatically by sampling the scan data, as described by JEONG *et al.* [JKHS02]. The method takes as input an unstructured point cloud that is obtained from range scans of an individual. A fully automated procedure is used to fit the generic face mesh of Section 5.1 to this point cloud, taking special care to match facial features such as ears and mouth. A hierarchical structure of displaced subdivision surfaces is then constructed on top of this control mesh, approximating the input geometry with increasing precision up to the sampling resolution of the input data. Figure 9.2 shows this construction process for one head model.

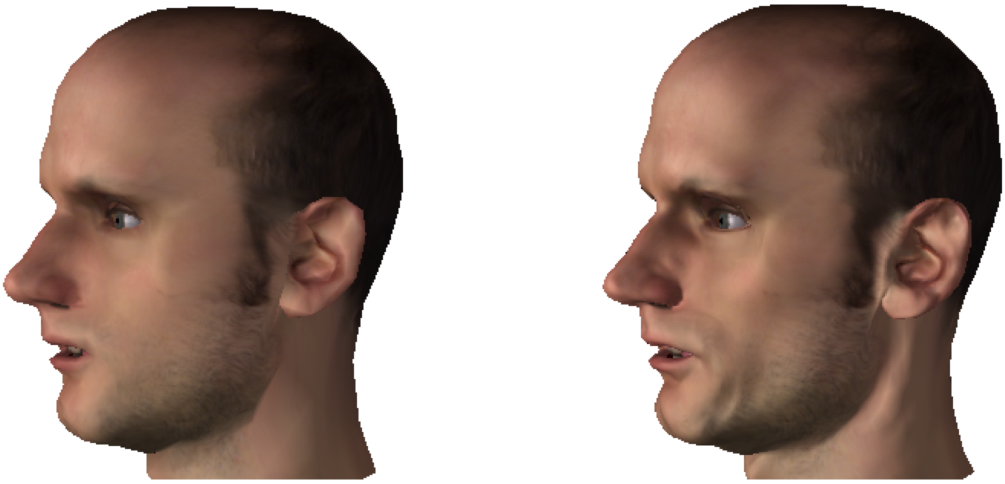


Figure 9.3: Head model rendered as standard triangle mesh (left, 2030 triangles) and displaced subdivision surface on that mesh (right, 2 refinement levels)

The subdivision surface in this case is constructed using the Modified Butterfly subdivision scheme [ZSS96]. Since this is an interpolating scheme, displacing a vertex of the subdivision control mesh corresponds directly to the local change of the limit surface. In the facial animation system, the coarse skin mesh serves as the control mesh for the subdivision surface, and can be efficiently animated. The level of detail for rendering the geometry is controlled separately by choosing the refinement level of the surface. For a given level, the support of a vertex in the control mesh is known [ISD02, ZSS97], and thus efficient animation can be achieved by only locally updating the refined mesh structure. In the current implementation, the algorithm does not achieve real-time frame rates, but delivers about twenty frames per second on a 1.7 GHz Xeon PC. A comparison of the standard, coarse face mesh rendered using plain OpenGL with a subdivided mesh is shown in Figure 9.3.

9.3 Dynamic Geometry Refinement

Instead of constructing a full subdivision surface, increased smoothness of the coarse triangle mesh for rendering can also be achieved by adding only local refinements. In my approach, refinements are dynamically introduced to the triangle mesh as it deforms. Triangles are split in a fashion similar to one or two steps of Loop subdivision [Loo87], but only in those areas where the mesh is highly deformed. Since these refinements are used for rendering only and can be easily computed on the fly, the original triangle mesh is not changed. This is advantageous for two reasons: first, the application doesn't have to deal with dynamic mesh connectivity, making integration into the existing system a simple plug-in operation. Second, the refined triangles are not retained between rendered frames, so additional memory usage is kept to a minimum. Figure 9.4 shows the integration into the facial animation system.

The method is based on an explicit reformulation of recursive adaptive refinement, which conceptually can be outlined as follows:

```
refine(region r):
  c := curvature(r)
  if (c > threshold)
    subdivide(r, c)
    for all sub-regions s in r
      refine(s)
  else
    draw(r)
```

Even if the tail recursion is flattened by transformation into a loop, two cost factors remain: the curvature has to be evaluated multiple times on the initial region (albeit on smaller and smaller parts), and the changes caused by a subdivision step have to be stored in the geometry before the subregions can be examined (or temporary storage must be allocated per subregion on each level of recursion).

By evaluating the curvature only once, these costs can be minimized. Based on the outcome, up to two refinements are performed in one step, thus eliminating the need for storing the altered geometry for further evaluation:

```
refine(region r):
  c := curvature(r)
  if (c > threshold)
    region s = subdivide1or2(r, c)
    draw(s)
  else
    draw(r)
```

In the implementation, the refinement procedure is applied to the triangle mesh just before rendering, as shown in Figure 9.4. Each edge of the given deformed mesh is examined to decide whether it should be split into two or more parts, causing subdivision of the adjacent triangles. It is assumed that the quality of the triangulation of the undeformed mesh is appropriate for visualization. Therefore an edge needs to be split only if the curvature of the surrounding mesh region has increased during mesh deformation. As a simple and efficient test, I use the dot product between the vertex normals at both ends of an edge. If this scalar value drops below the value that has been precomputed for the undeformed face geometry, there is more “bending” and the edge is marked for splitting once or twice (see Figure 9.5), depending on the difference of the dot products.

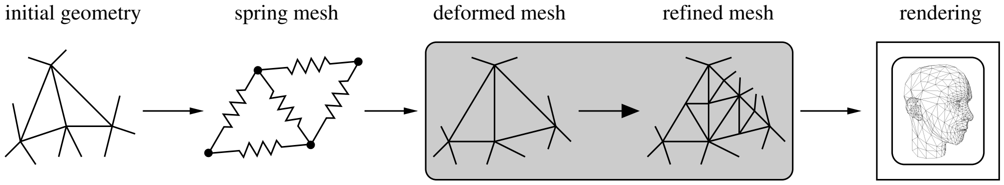


Figure 9.4: Overview of the physics-based facial animation system with dynamic refinement module. Without affecting the internal state of the simulation or the triangle mesh itself, refinements are computed from the deformed mesh and passed to the rendering engine.

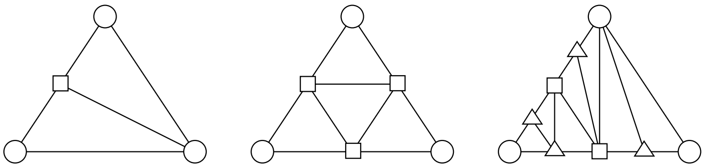


Figure 9.5: Examples for split configurations

The splitting criterion only uses the vertex normals of the existing nodes in the mesh. More complex criteria can be used as well, for instance, measuring discrete curvature on the mesh [Tau95]. The vertex normal dot product has proven to be sensitive to the kind of deformations that occur in facial animation (see Figure 9.7). Additionally, the criterion has the advantage of extremely low evaluation cost, provided that vertex normals have been computed before.

Using the new degrees of freedom provided by the split vertices, a smoother retriangulation approximating the input mesh is computed. For the smoothing, simple local rules are used that borrow from the subdivision idea [KHS01a, KHS02]. No new vertices are generated in the interior of an original triangle, thus avoiding evaluation of new interior edges and keeping the number of possible new triangulations manageable. The retriangulation is efficiently created by a table lookup operation, as shown in Figure 9.6. The resulting triangle set is then rendered instead of the original triangle, while unsplit triangles are rendered as usual. Since the computed refinements are not inserted into the input mesh, undoing refinements is not necessary. Also, no information about splits is retained between frames.

In practice, integration of dynamic refinement into the facial animation system results in no noticeable difference in performance. This is not surprising, since the computational load is mainly caused by the evaluation of the physics model and not by the rendering stage. Figure 9.7 shows a detail of the deformed mouth region during animation. The rendered mesh is visibly smoothed, with reduced shading artifacts and an improved silhouette of the opened mouth.

9.4 Discussion

The resolution of the original mesh is quite sufficient for facial animation in desktop applications, with the exception of well-defined facial wrinkles. Good results are obtained here using hardware-supported, computationally cheap shading techniques. For close-ups of highly deformable regions such as the mouth, local dynamic mesh refinement can increase the surface smoothness significantly, and thus has a use in these

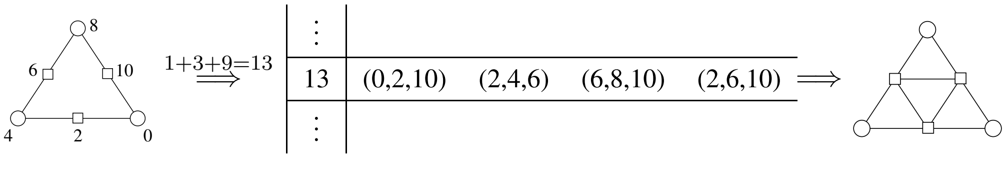


Figure 9.6: A split triangle is re-triangulated for rendering using a lookup table. Each split configuration corresponds to 1 of 27 possible combinations.

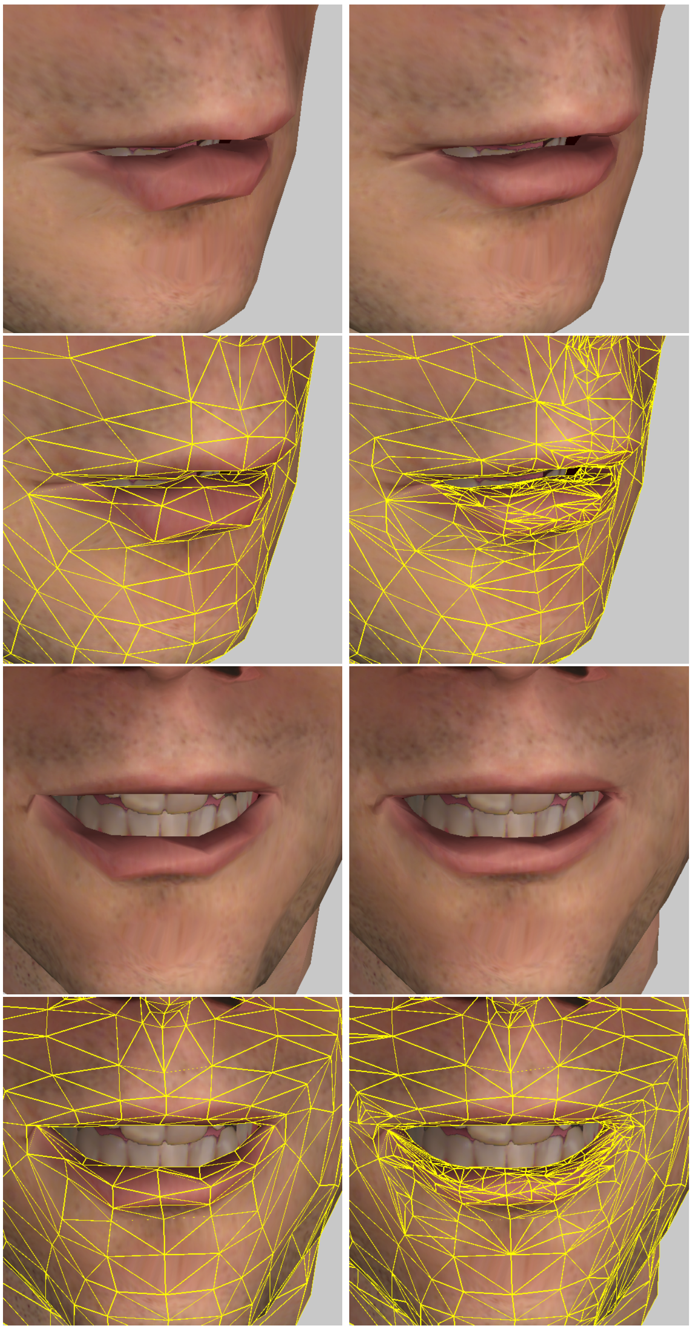


Figure 9.7: Snapshots from animation of a face mesh. Left: static mesh (2030 triangles). Right: dynamically refined mesh (2030-3120 triangles).

cases. This method has the additional advantages of very simple integration into the animation system and low computational overhead. For the future, the use of displaced subdivision surfaces probably has the most potential, especially once these surface types are supported in hardware. Currently, using subdivision surfaces implemented in software, the system becomes significantly slower and can as such not be used in real-time applications. Still, using displacements allows for geometric representation of very fine surface detail, which can be simulated only to a limited degree using hardware bump mapping and similar techniques. Also, by adjusting the refinement level of a subdivision surface, rendering performance and quality can be elegantly balanced. This enables adaptation to the needs of the application as well as to the computational power of the available hardware.

CHAPTER 10

Conclusion

Your face, my thane, is as a book, where men may read strange matters.
– William Shakespeare

In this dissertation, I have described a virtual head model with anatomical structure. The model is animated in a physics-based manner by use of muscle contractions that in turn cause skin deformations, where the simulation is efficient enough to achieve real-time frame rates on current PC hardware. Manual construction of such a structured head model is a difficult task, which is avoided in my approach by deriving new models from a prototype, employing a deformation method that reshapes the complete head structure. Without additional modeling tasks, this results in an immediately animatable virtual head. The general deformation method allows for several applications, first and foremost adaptation of a reference head model to individual scan data to produce animated head models of real persons. The deformation method is based on facial feature points, which leads to other interesting uses when an anthropometric standard set of facial landmarks is chosen: I have presented algorithms for simulation of human head growth and reconstruction of a face from a skull.

The facial modeling and animation system is used in teaching at the Max-Planck-Institut für Informatik and provides a solid foundation for further exploration of many related research areas, some of which are currently investigated in cooperation with other faculties of the Universität Saarbrücken and affiliated research institutions. Namely, these are the Institut für Computerlinguistik (speech animation) and the Institut für Rechtsmedizin (face reconstruction from skull data) at the Universität Saarbrücken, and the Deutsches Forschungszentrum für künstliche Intelligenz (emotional animation with synthesized speech).

The methods and techniques described are demonstrated on human head models in this dissertation. It is also feasible to use them for other types of creatures, with few modifications in the construction process. For a dragon head, such as shown in Figure 10.1, a human skull is certainly ill-suited to "real" dragon anatomy. A matching skull model can be hand-crafted, or created as an offset surface, obtained by cutting up and simplifying the input skin mesh. Laying out the muscles using the interactive tools described in Section 7.5, a prototype for the class of dragon models is constructed. This can then be used to generate a family of various dragon-shaped heads.

Since especially the animation part of this work is hard to reproduce in printed form, I'd like to point the reader to the official project web page situated at

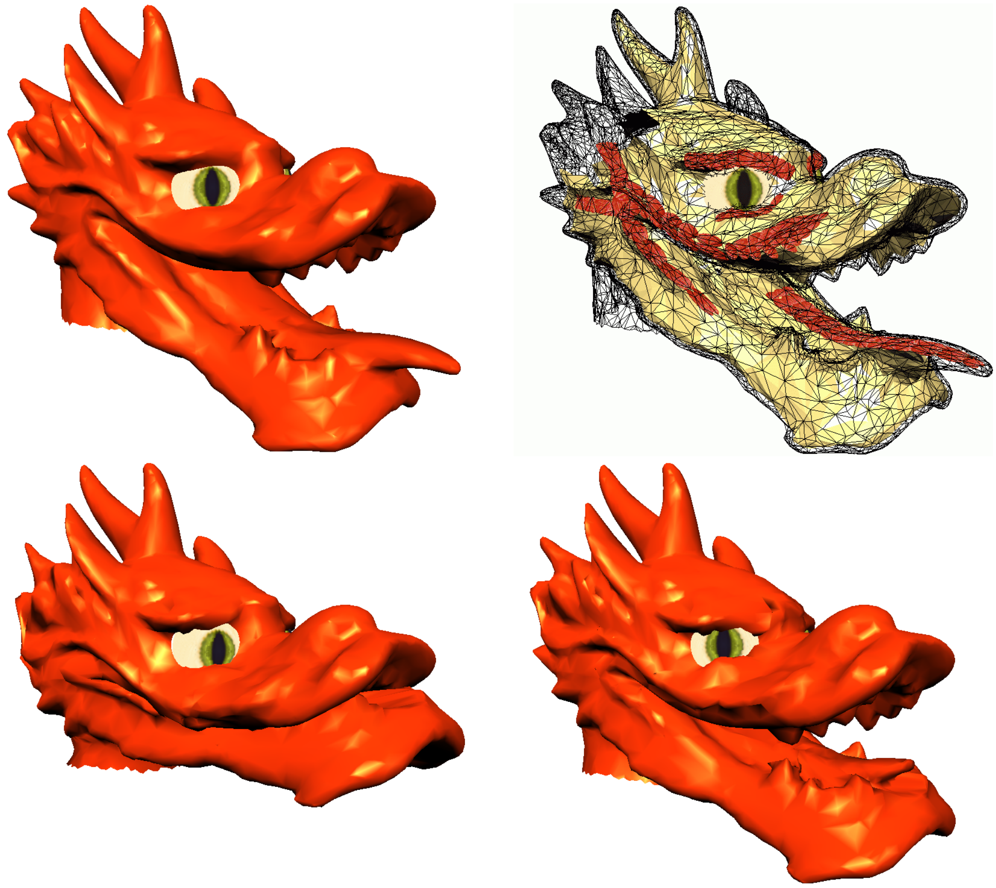


Figure 10.1: Construction of an animatable model from artificial head geometry. Clockwise from top left: Input mesh with eyes added; approximated skull and user-designed muscles; fierce expression; sad expression.

`http://www.mpi-sb.mpg.de/resources/FAM/`. In addition to current information, the site provides demonstration movie material covering many of the topics in this text.

10.1 Future Directions

Dealing with such complex subject matter as the shape and dynamics of the human face, I have by necessity focused on certain aspects of facial modeling and animation, and neglected others. Concentrating on the open problems, there are many opportunities for direct improvements as well as broader issues presenting challenging research topics. Judging from the experience with the current system, the most obvious areas for near-term future work are:

Skin simulation: The system is decidedly geared towards real-time animation and thus has a relatively simple skin model. Only elastic properties are included, and the changes in dynamic behavior of, for instance, old people's skin have not been investigated. Also, accurate formation of wrinkles in geometry is not included in the skin model. Effective formulation of a more complex skin structure requires efficient numerical simulation. Advanced integration schemes could pave the way for an improved skin model.

Collisions: Collisions of the skin with itself and other parts of the model are vital for reproduction of some effects, for instance, pressing the lips together or having (literally) the tongue in cheek. While collision *detection* mainly raises performance issues, proper *reaction* in a running simulation is a difficult problem.

Animation control: Lifelike facial motion has many nuances that are extremely difficult to put together by hand. The simple tracking mechanisms introduced in Chapter 5.3 already add substantially to the naturalness and credibility of speech animation. Refinement of the tracking techniques should allow for detailed analysis of the components of expressive animation, leading to insights that can be used to improve animation results. Animation experiments can be easily performed with the current system, allowing postulation and verification of working hypotheses. Some steps in this direction have already been taken by researchers [LBB02, LM99, TW90].

Facial components: The existing facial components eyes, teeth, and tongue are implemented using rather simple geometric entities. Especially the eye models deserve improvement with respect to a better integration into the face. From lifelong experience with looking at other peoples' eyes, any observer senses immediately when the eyes are not perfectly embedded or the eyelids do not smoothly blend into the facial skin. An important part of the head model that is completely missing so far is hair. This is a vast topic that has it's own dedicated research community within computer graphics [MTHK00].

Rendering: In this work, the skin shading is essentially restricted to texturing with hardware-accelerated bump mapping. For highly realistic rendering of facial skin, effects such as self-shadowing, subsurface scattering, and anisotropic reflection properties of skin must be taken into account. High-quality rendering algorithms have traditionally been too expensive for real-time applications, but the rapid progress in graphics hardware promises a steep increase in achievable realism in this area in the near future.

In a broader view, one of the recurring problems in building any discrete model of a real-world object is the use of approximations and unknown variables. The astonishing complexity in detail that one finds when taking a closer look at the human face seems to render any attempt at realistic simulation futile. Another limiting factor is the large amount of unknowns in facial anatomy and dynamics. Current medical knowledge does not allow detailed, complete specification of the changes in a moving face, for instance, the exact contraction behavior of a muscle. Given this seemingly desolate situation, I still hope to have demonstrated with this work how even a system working under real-time constraints can profit from a model built on close examination of the real world. With the current state of physics-based facial animation, maybe the most important and interesting question at this point is: *which model improvements have a significant visual impact?* For instance, is it reasonable to extend a muscle model to include advanced dynamics, or will the result simply not be visible? An answer to this question requires extensive studies of all aspects of the system from anatomical modeling, over numerical simulation, to rendering and perceptional issues.

10.2 Outlook

Thinking further, a realistic head model is just one, albeit vital, part of a believable virtual character. The combination of head and body is a logical progression, but again a non-trivial task: for instance, body language and facial language are closely related and must be choreographed properly to achieve a believable personality. Interaction between body parts and the head (such as wiping one's brow) must be supported. Once a virtual character is established, communication between multiple characters can be modeled, enabling social activities and virtual societies. In the realm of human communication, trust has a key role, and only truly believable characters will be considered trustworthy. Since most essential communication in daily life is face to face, we come full circle at this point: realistic facial modeling and animation is a cornerstone for real, natural communication in a virtual space. The topic will continue to provide many research challenges for quite some time into the future, but as the saying goes: "The joy is in the journey."

Bibliography

[A$^+$01] A. Adamson et al. Shrek: The Story Behind the Screen. In *SIGGRAPH 2001 Course Notes*, 2001.

[AHK$^+$02] I. Albrecht, J. Haber, K. Kähler, M. Schröder, and H.-P. Seidel. "May I talk to you? :-)"– Facial Animation from Text. In *Proc. Pacific Graphics 2002*, pages 77–86. IEEE Computer Society Press, October 2002.

[AHS02a] I. Albrecht, J. Haber, and H.-P. Seidel. Automatic Generation of Non-Verbal Facial Expressions from Speech. In *Proc. Computer Graphics International 2002*, pages 283–293, 2002.

[AHS02b] I. Albrecht, J. Haber, and H.-P. Seidel. Speech Synchronization for Physics-based Facial Animation. In *Proc. WSCG 2002*, pages 9–16, February 2002.

[AHTN01] Y. Aoki, S. Hashimoto, M. Terajima, and A. Nakasima. Simulation of Postoperative 3D Facial Morphology using a Physics-Based Head Model. *The Visual Computer*, 17(2):121–131, 2001.

[Alb01] I. Albrecht. Speech Synchronization for Physics-Based Human Face Models. Master's thesis, Universität des Saarlandes, Saarbrücken, 2001.

[All90] E. Allen. Computerized photo aging and the search for missing children. *Interpol International Criminal Police Review*, September-October 1990.

[Arc97] K. M. Archer. Craniofacial Reconstruction using hierarchical B-Spline Interpolation. Master's thesis, University of British Columbia, Department of Electrical and Computer Engineering, August 1997.

[Bat82] K. J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[BB02] M. Byun and N. I. Badler. FacEMOTE: Qualitative Parametric Modifiers for Facial Animations. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 65–72. ACM SIGGRAPH, July 2002.

[BCS97] C. Bregler, M. Covell, and M. Slaney. Video Rewrite: Driving Visual Speech with Audio. In *Proc. ACM SIGGRAPH 1997*, Computer Graphics Proceedings, Annual Conference Series, pages 353–360. ACM Press / ACM SIGGRAPH, August 1997.

[BHG92] D. E. Breen, D. H. House, and P. H. Getto. A Physically-Based Particle Model of Woven Cloth. *The Visual Computer*, 8(5-6):264–277, June 1992.

[BN92] T. Beier and S. Neely. Feature-based image metamorphosis. In *Computer Graphics (Proc. ACM SIGGRAPH 92)*, volume 26, pages 35–42, July 1992.

[Boo97a] F. L. Bookstein. *Morphometric Tools for Landmark Data*. Cambridge University Press, 1997.

[Boo97b] F. L. Bookstein. Shape and the Information in Medical Images: A Decade of the Morphometric Synthesis. *Computer Vision and Image Understanding*, 66(2):97–118, 1997.

[Bra99] M. Brand. Voice Puppetry. In *Proc. ACM SIGGRAPH 1999*, Computer Graphics Proceedings, Annual Conference Series, pages 21–28. ACM Press / ACM SIGGRAPH, August 1999.

[BS83] N. M. Brooke and Q. Summerfield. Analysis, synthesis, and perception of visible articulatory movements. *Journal of Phonetics*, 11:63–76, 1983.

[BV99] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Proc. ACM SIGGRAPH 1999*, Computer Graphics Proceedings, Annual Conference Series, pages 187–194. ACM Press / ACM SIGGRAPH, August 1999.

[BW97] Body Works: A 3D Journey Through the Human Anatomy. Software CD-ROM, The Learning Company, 1997.

[BW98] D. Baraff and A. Witkin. Large Steps in Cloth Simulation. In *Proc. ACM SIGGRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 43–54. ACM Press / ACM SIGGRAPH, July 1998.

[CB00] J. Cassell and T. Bickmore. External manifestations of trustworthiness in the interface. *Communications of the ACM*, 43(12):50–56, 2000.

[CBB$^+$99] J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. Embodiment in conversational interfaces: Rea. In *Proc. CHI 99 conference on Human factors in computing systems : the CHI is the limit*, pages 520–527. ACM Press, 1999.

[CBC$^+$01] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. Richard Fright, B. C. McCallum, and T. R. Evans. Reconstruction and Representation of 3D Objects With Radial Basis Functions. Computer Graphics Proceedings, Annual Conference Series, pages 67–76. ACM Press / ACM SIGGRAPH, August 2001.

[CHP89] J. E. Chadwick, D. R. Haumann, and R. E. Parent. Layered Construction for Deformable Animated Characters. In *Computer Graphics (Proc. ACM SIGGRAPH 89)*, pages 243–252, July 1989.

[CKS98] S. Campagna, L. Kobbelt, and H.-P. Seidel. Directed Edges: A Scalable Representation for Triangle Meshes. *Journal of Graphics Tools*, 3(4):1–11, 1998.

[CM93] M. M. Cohen and D. W. Massaro. Modeling Coarticulation in Synthetic Visual Speech. In *Models and Techniques in Computer Animation*, pages 139–156. Springer, 1993.

[CMS98] P. Cignoni, C. Montani, and R. Scopigno. A Comparison of Mesh Simplification Algorithms. *Computers & Graphics*, 22(1):37–54, 1998.

[CW92] J. M. Cychosz and W. N. Waggenspeck Jr. *Intersecting a ray with a quadric surface*, volume III of *Graphics Gems*, pages 275–283. Academic Press, 1992.

[CZ92] D. T. Chen and D. Zeltzer. Pump it up: Computer Animation of a Biomechanically Based Model of Muscle using the Finite Element Method. In *Proc. ACM SIGGRAPH 1992*, Computer Graphics Proceedings, Annual Conference Series, pages 89–98. ACM Press / ACM SIGGRAPH, July 1992.

[DCH88] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume Rendering. In *Computer Graphics (Proc. ACM SIGGRAPH 88)*, volume 22, pages 65–74, 1988.

[Den88] X. Q. Deng. *A Finite Element Analysis of Surgery of the Human Facial Tissue*. PhD thesis, Columbia University, New York, 1988.

[DHT$^+$00] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proc. ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 145–156. ACM Press / ACM SIGGRAPH, July 2000.

[DKT98] T. DeRose, M. Kass, and T. Truong. Subdivision Surfaces in Character Animation. In *Proc. ACM SIGGRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 85–94. ACM Press / ACM SIGGRAPH, July 1998.

[DMS98] D. DeCarlo, D. Metaxas, and M. Stone. An Anthropometric Face Model using Variational Techniques. In *Proc. ACM SIGGRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 67–74. ACM Press / ACM SIGGRAPH, July 1998.

[Duc77] J. Duchon. Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics*, pages 85–100, 1977.

[Duc90] F. A. Duck. *Physical Properties of Tissue: A Comprehensive Reference Book*. Academic, London, 1990.

[EF77] P. Ekman and W. V. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Palo Alto, 1977.

[EGP02] T. Ezzat, G. Geiger, and T. Poggio. Trainable Videorealistic Speech Animation. *ACM Transactions on Graphics*, 21(3):388–398, July 2002.

[EN72] A. El-Nofely. Anthropometric study of growth changes of some head and face measurements in an egyptian group. *Egyptian Dental J.*, 18(2):141–150, 1972.

[EP97] I. Essa and A. Pentland. Coding, Analysis, Interpretation and Recognition of Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[EPMT98] M. Escher, I. Pandzic, and N. Magnenat-Thalmann. Facial Deformations for MPEG-4. *Computer Animation '98*, June 1998. Philadelphia, Pennsylvania, USA.

[Far94] L. G. Farkas, editor. *Anthropometry of the Head and Face*. Raven Press, 2nd edition, 1994.

[FN93] B. A. Fedosyutkin and J. V. Nainys. *Forensic Analysis of the Skull*, chapter 15: The Relationship of Skull Morphology to Facial Features, pages 199–213. Wiley-Liss, 1993.

[FPH92] L. G. Farkas, J. C. Posnick, and T. M. Hreczko. Growth patterns of the face: A morphometric study. *Cleft Palate-Craniofacial J.*, 29(4):308–315, 1992.

[Fun93] Y. C. Fung. *Biomechanics: Mechanical Properties of Living Tissues*. Springer, New York, Inc., 1993.

[GGW$^+$98] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making Faces. In *Proc. ACM SIGGRAPH 1998*, Computer Graphics Proceedings, Annual Conference Series, pages 55–66. ACM Press / ACM SIGGRAPH, July 1998.

[GH97] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, pages 209–216, 1997.

[GMTA$^+$96] T. Guiard-Marigny, N. Tsingos, A. Adjoudani, C. Benoit, and M. P. Gascuel. 3D Models of the lips for realistic speech animation. In *Computer Animation '96*, May 1996.

[Grü93] Oskar Grüner. *Forensic Analysis of the Skull*, chapter 3: Identification of Skulls: A Historical Review and Practical Applications. Wiley-Liss, 1993.

[GSD90] J. S. Genecov, P. M. Sinclair, and P. C. Dechow. Development of the nose and soft tissue profile. *Angle Orthodontist*, 60(3):191–198, 1990.

[GVSS00] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal Meshes. In *Proc. ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series. ACM Press / ACM SIGGRAPH, July 2000.

[HE01] M. Hauth and O. Etzmuss. A High Performance Solver for the Animation of Deformable Objects using Advanced Numerical Methods. In *Computer Graphics Forum*, volume 20, pages 319–328, 2001.

[HHK92] W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct Manipulation of Free-Form Deformations. In *Proc. ACM SIGGRAPH 1992*, volume 26 of *Computer Graphics Proceedings, Annual Conference Series*, pages 177–184. ACM Press / ACM SIGGRAPH, July 1992.

[HKA$^+$01] J. Haber, K. Kähler, I. Albrecht, H. Yamauchi, and H.-P. Seidel. Face to Face: From Real Humans to Realistic Facial Animation. In *Proc. Israel-Korea Binational Conference on Geometrical Modeling and Computer Graphics*, pages 73–82, October 2001.

[HPW88] D. Hill, A. Pearce, and B. Wyvill. Animating speech: an automated approach using speech synthesised by rules. *The Visual Computer*, 3:277–287, 1988.

[HRPM93] R. P. Helmer, S. Röhricht, D. Petersen, and F. Möhr. *Forensic Analysis of the Skull*, chapter 17: Assessment of the Reliability of Facial Reconstruction, pages 229–246. Wiley-Liss, 1993.

[ISD02] I. P. Ivrissimtzis, M. A. Sabin, and N. Dodgson. On the support of recursive subdivision. Technical Report UCAM-CL-TR-544, Cambridge University, Cambridge, UK, September 2002.

[IYYT93] T. Ishii, T. Yasuda, S. Yokoi, and J. Toriwaki. A Generation Model for Human Skin Texture. In *Proc. Computer Graphics International '93*, pages 139–150, 1993.

[JKHS02] W.-K. Jeong, K. Kähler, J. Haber, and H.-P. Seidel. Automatic Generation of Subdivision Surface Head Models from Point Cloud Data. In *Proc. Graphics Interface 2002*, pages 181–188. A K Peters, May 2002.

[KBB$^+$00] L. Kobbelt, S. Bischoff, M. Botsch, K. Kähler, C. Rössl, R. Schneider, and J. Vorsatz. Geometric Modeling Based on Polygonal Meshes. In *Eurographics 2000 Tutorial Notes*, 2000.

[KGC$^+$96] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser, and Y. I. H. Parish. Simulating Facial Surgery using Finite Element Methods. In *Proc. ACM SIGGRAPH 1996*, Computer Graphics Proceedings, Annual Conference Series, pages 421–428. ACM Press / ACM SIGGRAPH, 1996.

[KGPG96] E. Keeve, S. Girod, P. Pfeifle, and B. Girod. Anatomy-Based Facial Tissue Modeling Using the Finite Element Method. In *IEEE Visualization '96*, pages 21–28. IEEE Computer Society Press, October 1996.

[KHS01a] K. Kähler, J. Haber, and H.-P. Seidel. Dynamic Refinement of Deformable Triangle Meshes for Rendering. In *Proc. Computer Graphics International 2001*, pages 285–290, 2001.

[KHS01b] K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based Muscle Modeling for Facial Animation. In *Proc. Graphics Interface 2001*, pages 37–46, 2001.

[KHS02] K. Kähler, J. Haber, and H.-P. Seidel. Dynamically refining animated triangle meshes for rendering. *The Visual Computer*, 19(5):310–318, 2002.

[KHS03] K. Kähler, J. Haber, and H.-P. Seidel. Reanimating the Dead: Reconstruction of Expressive Faces from Skull Data. *ACM Transactions on Graphics*, pages 564–561, July 2003.

[KHYS02] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: Generating animated head models with anatomical structure. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 55–64. ACM SIGGRAPH, July 2002.

[KLP76] W. Kahle, H. Leonhardt, and W. Platzer. *Taschenatlas der Anatomie*. Thieme, 1976.

[KMMTT91] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. SMILE: a Multilayered Facial Animation System. In T. L. Kunii, editor, *Modeling in Computer Graphics*. Springer, 1991.

[Koe02] R. Koenen, editor. Overview of the MPEG-4 Standard. `http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm`, March 2002.

[KP00] S. A. King and R. E. Parent. A Parametric Tongue Model for Animated Speech. In *Proc. Computer Animation and Simulation 2000*, pages 3–13, 2000.

[KPO00] S. A. King, R. E. Parent, and B. Olsafsky. An Anatomically-Based 3D Parametric Lip Model to Support Facial Animation and Synchronized Speech. In *Proc. Deform 2000*, pages 7–19, 2000.

[Kro46] W. M. Krogman. The reconstruction of the living head from the skull. *FBI Law Enforcement Bulletin*, July 1946.

[KS98] J. Keener and J. Sneyd. *Mathematical Physiology*, chapter 18. Springer, 1998.

[Lai00] J. Lai, editor. Conversational interfaces. *Communications of the ACM*, 43(9):24–73, 2000.

[Las87] J. Lasseter. Principles of traditional animation applied to 3D computer animation. In *Computer Graphics (Proc. ACM SIGGRAPH 87)*, pages 35–44, 1987.

[Las95] J. Lasseter, director. Toy Story. DVD, Walt Disney Home Video, 1995. ASIN: B000059XUT.

[LBB02] S. P. Lee, J. B. Badler, and N. I. Badler. Eyes Alive. *ACM Transactions on Graphics*, 21(3):637–644, July 2002.

[LBV93] G. V. Lebedinskaya, T. S. Balueva, and E. V. Veselovskaya. *Forensic Analysis of the Skull*, chapter 14: Principles of Facial Reconstruction, pages 183–198. Wiley-Liss, 1993.

[LM99] J. C. Lucero and K. G. Munhall. A model of facial biomechanics for speech production. *J. Acoust. Soc. Am.*, 106(5):2834–2842, November 1999.

[LMH00] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proc. ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series. ACM Press / ACM SIGGRAPH, 2000.

[LMT00] W.-S. Lee and N. Magnenat-Thalmann. Fast Head Modeling for Animation. *Image and Vision Computing*, 18(4):355–364, 2000.

[Loo87] C. T. Loop. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[LP87] J. P. Lewis and F. I. Parke. Automated lipsynch and speech synthesis for character animation. In *Proc. Human Factors in Computing Systems and Graphics Interface '87*, pages 143–147, 1987.

[LSZ01] Z. Liu, Y. Shan, and Z. Zhang. Expressive Expression Mapping with Ratio Images. In *Proc. ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 271–276. ACM Press / ACM SIGGRAPH, 2001.

[LTC99] A. Lanitis, C. Taylor, and T. Cootes. Modeling the Process of Ageing in Face Images. In *Proc. 7th IEEE International Conference on Computer Vision*, volume I, pages 131–136. IEEE Computer Society Press, 1999.

[LTW95] Y. Lee, D. Terzopoulos, and K. Waters. Realistic Modeling for Facial Animations. In *Proc. ACM SIGGRAPH 1995*, Computer Graphics Proceedings, Annual Conference Series, pages 55–62. ACM Press / ACM SIGGRAPH, August 1995.

[LWCS96] S. Lee, G. Wolberg, K.-Y. Chwa, and S. Y. Shin. Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2(4), December 1996. ISSN 1077-2626.

[LWMT99] W.-S. Lee, Y. Wu, and N. Magnenat-Thalmann. Cloning and Aging in a VR Family. In *IEEE Virtual Reality*, March 1999.

[MC96] S. Michael and M. Chen. The 3D reconstruction of facial features using volume distortion. In *Proc. 14th Eurographics UK Conference*, pages 297–305, 1996.

[MGR00] S. Marschner, B. Guenter, and S. Raghupathy. Modeling and Rendering for Realistic Facial Animation. In *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 231–242. Eurographics, 2000.

[ML74] W. J. Moore and C. L. B. Lavelle. *Growth of the Facial Skeleton in the Hominoidea*. Academic Press, London, 1974.

[MMT97] L. Moccozet and N. Magnenat-Thalmann. Dirichlet Free-Form Deformations and Their Application to Hand Simulation. *Computer Animation '97*, June 1997.

[MTHK00] N. Magnenat-Thalmann, S. Hadap, and P. Kalra. State of the Art in Hair Simulation. In *International Workshop on Human Modeling and Animation*, 2000.

[MWL+99] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-based BRDF Measurement Including Human Skin. In *Eurographics Rendering Workshop 1999*. Springer Wein / Eurographics, June 1999.

[MYIS95] S. Miyasaka, M. Yoshino, K. Imaizumi, and S. Seta. The computer-aided facial reconstruction system. *Forensic Science Int.*, 74(1-2):155–165, 1995.

[NN01] J. Noh and U. Neumann. Expression Cloning. In *ACM Transactions on Graphics2001*, pages 277–288, July 2001.

[Par72] F. I. Parke. Computer generated animation of faces. In *Proceedings of the ACM annual conference*, pages 451–457, 1972.

[Par74] F. I. Parke. A Parametric Model for Human Faces. Technical report, University of Utah, Salt Lake City, Utah, 1974.

[Par82] F. I. Parke. Parameterized Models for Facial Animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, November 1982.

[PB81] S. M. Platt and N. I. Badler. Animating Facial Expressions. In *Computer Graphics (Proc. ACM SIGGRAPH 81)*, pages 245–252, August 1981.

[Pel91] C. Pelachaud. *Communication and Coarticulation in Facial Animation*. PhD thesis, University of Pennsylvania, 1991.

[PHL+98] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing Realistic Facial Expressions from Photographs. In *Proc. ACM SIGGRAPH 1998*, Computer Graphics Proceedings, Annual

Conference Series, pages 75–84. ACM Press / ACM SIGGRAPH, July 1998.

[PRZ92] S. Pieper, J. Rosen, and D. Zeltzer. Interactive graphics for plastic surgery: A task-level analysis and implementation. In *1992 Symposium on Interactive 3D Graphics*, volume 25, pages 127–134, March 1992.

[PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, Massachusetts, 2nd edition, 1992.

[PW96] F. I. Parke and K. Waters, editors. *Computer Facial Animation*. A K Peters, Wellesley, Massachusetts, 1996.

[PWWH86] A. Pearce, B. Wyvill, G. Wyvill, and D. Hill. Speech and expression: A computer solution to face animation. In *Proc. Graphics Interface '86*, pages 136–140, May 1986.

[Ras78] G. Rassner. *Atlas der Dermatologie und Venerologie*. Urban & Schwarzenberg, 1978.

[RC80] J. S. Rhine and H. R. Campbell. Thickness of facial tissues in American blacks. *Journal of Forensic Sciences*, 25(4):847–858, 1980.

[RM84] J. S. Rhine and C. E. Moore. Tables of facial tissue thickness of American Caucasoids in forensic anthropology. *Maxwell Museum Technical Series*, 1, 1984.

[Rus92] P. Russell. Image processing offers hope in the search for missing children. *Advanced Imaging*, 1992.

[SF00] A. Szunyoghy and G. Fehér. *Menschliche Anatomie für Künstler*. Könemann, Köln, 2000.

[SM01] H. Sakaguchi and M. Sakakibara, directors. Final Fantasy: The Spirits Within. DVD, Special Edition, Columbia Tri-Star, 2001. ASIN: B00003CY5D.

[SMT00] H. Seo and N. Magnenat-Thalmann. LoD Management on Animating Face Models. In *Proc. IEEE Virtual Reality 2000*, pages 161–168, 2000.

[Sob01] J. Sobotta. *Atlas of Human Anatomy, Volume 1: Head, Neck, Upper Limb*. Lippincot Williams & Wilkins, 13th english / english edition, 2001.

[SP86] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics (Proc. ACM SIGGRAPH 86)*, 20(4):151–160, August 1986.

[SPCM97] F. Scheepers, R. E. Parent, W. E. Carlson, and S. F. May. Anatomy-Based Modeling of the Human Musculature. In *Proc. ACM SIGGRAPH 1997*, Computer Graphics Proceedings, Annual Conference Series, pages 163–172. ACM Press / ACM SIGGRAPH, August 1997.

[Spe98] E.-J. Speckmann. *Bau und Funktionen des menschlichen Körpers. Praxisorientierte Anatomie und Physiologie*. Urban & Fischer, 19th edition, 1998.

[Ste79] G. K. Steigleider. *Dermatologie und Venerologie*. Thieme, 1979.

[Ste00] N. Stephenson. *Snow Crash*. Bantam, reprint edition, 2000.

[SWV02] M. Simmons, J. Wilhelms, and A. Van Gelder. Model-Based Reconstruction for Creature Animation. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 139–146, July 2002.

[Tan62] J. M. Tanner. *Growth at Adolescence*. Blackwell, Oxford, 1962.

[Tan88] J. M. Tanner. *Human Biology*, chapter Human growth and constitution, pages 337–435. Oxford University Press, 1988.

[Tau95] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral. In *Proc. International Conference on Computer Vision*, pages 902–907, 1995.

[Tay01] K. T. Taylor. *Forensic Art and Illustration*. CRC Press LLC, 2001.

[TBP01] B. Tiddeman, M. Burt, and D. Perrett. Prototyping and Transforming Facial Textures for Perception Research. *IEEE Computer Graphics and Applications*, 21(5):42–50, September / October 2001.

[TCRS00] M. Tarini, P. Cignoni, C. Rocchini, and R. Scopigno. Real Time, Accurate, Multi-Featured Rendering of Bump Mapped Surfaces. *Computer Graphics Forum*, 19(3), August 2000.

[TG98] R. Turner and E. Gobbetti. Interactive construction and animation of layered elastically deformable characters. *Computer Graphics Forum*, 17(2):135–152, 1998.

[TJ81] F. Thomas and O. Johnston. *Disney Animation: The Illusion of Life*. Walt Disney Productions, New York, 1981.

[TLSP80] J. T. Todd, S. M. Leonard, R. E. Shaw, and J. B. Pittenger. The perception of human growth. *Scientific American*, (242):106–114, 1980.

[TW90] D. Terzopoulos and K. Waters. Physically-based Facial Modelling, Analysis, and Animation. *Journal of Visualization and Computer Animation*, 1(2):73–80, December 1990.

[TYHS02] M. Tarini, H. Yamauchi, J. Haber, and H.-P. Seidel. Texturing Faces. In *Proc. Graphics Interface 2002*, pages 89–98. A K Peters, May 2002.

[Van98] A. Van Gelder. Approximate Simulation of Elastic Membranes by Triangulated Spring Meshes. *Journal of Graphics Tools*, 3(2):21–41, 1998.

[Ver67] L. Verlet. Computer experiments on classical fluids: Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159(1):98–103, 1967.

[Ver90] M. Vercauteren. Age effects and secular trend in a cross-sectional sample: application to four head dimensions in Belgian adults. *Hum. Biol.*, 62(5):681–688, 1990.

[Ves94] F. J. Vesely. *Computational Physics: An Introduction*. Plenum Press, New York, 1994.

[VMT01] P. Volino and N. Magnenat-Thalmann. Comparing efficiency of integration methods for cloth simulation. In *Proc. Computer Graphics International 2001*, pages 265–272, July 2001.

[VVMN00] P. Vanezis, M. Vanezis, G. McCombe, and T. Niblett. Facial reconstruction using 3-D computer graphics. *Forensic Science Int.*, 108(2):81–95, 2000.

[VY92] M.-L. Viaud and H. Yahia. Facial Animation with Wrinkles. In *Computer Animation and Simulation '92*, September 1992.

[Wat87] K. Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics (Proc. ACM SIGGRAPH 87)*, pages 17–24, July 1987.

[WF95] K. Waters and J. Frisbie. A Coordinated Muscle Model for Speech Animation. In *Proc. Graphics Interface '95*, pages 163–170, May 1995.

[WH86] W. A. Weijs and B. Hillen. Correlations between the cross-sectional area of the jaw muscles and craniofacial size and shape. *Am. J. Phys. Anthropol.*, 70:423–431, 1986.

[WKMMT99] Y. Wu, P. Kalra, L. Moccozet, and N. Magnenat-Thalmann. Simulating Wrinkles and Skin Aging. *The Visual Computer*, 15(4):183–198, 1999.

[WL93] K. Waters and T. M. Levergood. DECface: An Automatic Lip-Synchronization Algorithm for Synthetic Faces. Technical Report CRL 93/4, Digital Equipment Corporation, Cambridge Research Lab, 1993.

[WMTT94] Y. Wu, N. Magnenat-Thalmann, and D. Thalmann. A Plastic-Visco-Elastic Model for Wrinkles in Facial Animation and Skin Aging. In *Proc. Pacific Graphics '94*, pages 201–214, August 1994.

[WND96] M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide*. Addison–Wesley, 2nd edition, 1996.

[WT91] K. Waters and D. Terzopoulos. Modeling and Animating Faces Using Scanned Data. *Journal of Visualization and Computer Animation*, 2(4):123–128, October–December 1991.

[WV97] J. Wilhelms and A. Van Gelder. Anatomically Based Modeling. In *Proc. ACM SIGGRAPH 1997*, Computer Graphics Proceedings, Annual Conference Series, pages 173–180. ACM Press / ACM SIGGRAPH, August 1997.

[WW92] W. Welch and A. Witkin. Variational surface modeling. In *Computer Graphics (Proc. ACM SIGGRAPH 92)*, volume 26, pages 157–166, July 1992.

[yI93] G. J. y'Edynak and M. Y. İşcan. *Forensic Analysis of the Skull*, chapter 16: Anatomical and Artistic Guidelines for Forensic Facial Reconstruction, pages 215–227. Wiley-Liss, 1993.

[ZPM98] C. P. E. Zollikofer, M. S. Ponce De León, and R. D. Martin. Computer-Assisted Paleoanthropology. *Evolutionary Anthropology*, 6(2):41–54, 1998.

[ZS00] D. Zorin and P. Schröder. Subdivision for Modeling and Animation. In *SIGGRAPH 2000 Course Notes*, 2000.

[ZSS96] D. Zorin, P. Schröder, and W. Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology. In *Proc. ACM SIGGRAPH 1996*, Computer Graphics Proceedings, Annual Conference Series, pages 189–192. ACM Press / ACM SIGGRAPH, 1996.

[ZSS97] D. Zorin, P. Schröder, and W. Sweldens. Interactive Multiresolution Mesh Editing. In *Proc. ACM SIGGRAPH 1997*, Computer Graphics Proceedings, Annual Conference Series, pages 259–268. ACM Press / ACM SIGGRAPH, August 1997.

APPENDIX A

Glossary of Medical Terms

This appendix lists anatomical, anthropometric, and a few general medical terms that are used in the main text. For some terms that have a broader general meaning, here the explanation most closely related to head anatomy is given. Names of facial muscles are only included in the form of common synonyms to the name used in this text. A complete description of the muscles referenced in this dissertation can be found in Chapter 4. A list of names for facial landmarks is given in Figure 8.1 (Section 8.2).

anatomy	the study of structures of the body and their relationships
anterior	in front
aponeurosis	any of the deeper and thicker →fascia that attach muscles to bones; similar to a flattened tendon
auricle	the external ear
caninus	*syn.* levator anguli oris
cartilage	translucent, elastic tissue; *syn.* gristle
cephalic	pertaining to the head
collagen	protein found in →connective tissue, for instance in the skin
compressor naris	*syn.* nasalis
connective tissue	tissue that holds different structures together
corium	*syn.* →dermis
coronal plane	*syn.* →frontal plane
cranial	pertaining to the skull
depressor anguli oris	*syn.* triangularis
dermal	related to the dermis
dermis	*syn.* →corium; layer of the skin lying immediately below the epidermis; divided into the papillary layer and the →reticular layer
elastin	a protein that is flexible and elastic when moist, but brittle when dry
epicranial aponeurosis	*syn.* galea aponeurotica

epidermis	top layer of the skin, composed of several strata (→stratum)
epithelium, epithelial tissue	closely aggregated, tightly adherent cells with very little substance between them; commonly found on the surfaces of the body and organs
fascia	a band or sheet of fibrous tissue which connects muscles and skin
Frankfort Horizontal	standard position for anthropometric measurements: the line connecting the landmarks *or* (orbitale) and *po* (porion) is maintained horizontal
frontal plane	plane that lies at right angles to both the →median plane and the →transverse plane; divides the head into front and back parts
ground substance	→matrix; amorphous gel-like substance that fills the space between the elements of the →dermis
helix	the incurved margin or rim of the external ear
horizontal plane	*syn.* →transverse plane
hypodermis	*syn.* →subcutis
keratin	a protein that is the primary component of the →epidermis
lateral	away from the middle, towards the outside
malar bone	*syn.* →zygomatic bone
matrix	intercellular material
medial	towards the middle, or inside
median plane	vertical plane that divides the head into left and right halves
melanin	black pigment found in the pigment-bearing cells of the skin
mucous membrane	membrane which lines passages and cavities which communicate with the exterior, habitually secreting →mucus
mucus	a viscid fluid secreted by →mucous membranes, which serves to moisten and protect
perioral	in the area around the mouth
physiology	the study of the functions of the body structures
posterior	behind, towards the rear
quadratus labii inferioris	*syn.* depressor labii inferioris
reticular	forming a network
reticular layer	layer of dense connective tissue forming the deeper part of the →dermis

sagittal plane	any plane parallel to the →median plane
sebaceous gland	small →subcutaneous gland, which secretes an oily semifluid matter, composed mostly of fat
sebum	fatty substance, secreted from a →sebaceous gland
septum	dividing partition between two tissues or cavities
squamous	covered with, or consisting of, scales; scale-like
stratified	layered
stratum	layer of the skin; each stratum has different functions
subcutis	layer of connective tissue beneath the dermis
subcutaneous	beneath the skin
supraorbital ridge	*syn.* superciliary arch
tendinous	pertaining to a tendon
transverse plane	*syn.* horizontal plane, at right angles with the →median plane; divides the head into upper and lower parts
vertex	the top, or crown, of the head

APPENDIX B

Publications

The following list contains all publications that have resulted from the development of the facial modeling and animation system presented in this dissertation, either as part of my own research or building on it.

[1] I. Albrecht, J. Haber, K. Kähler, M. Schröder, and H.-P. Seidel. ”May I talk to you? :-)”– Facial Animation from Text. In *Proc. Pacific Graphics 2002*, pages 77–86. IEEE Computer Society Press, October 2002.

[2] I. Albrecht, J. Haber, and H.-P. Seidel. Automatic Generation of Non-Verbal Facial Expressions from Speech. In *Proc. Computer Graphics International 2002*, pages 283–293, 2002.

[3] I. Albrecht, J. Haber, and H.-P. Seidel. Speech Synchronization for Physics-based Facial Animation. In *Proc. WSCG 2002*, pages 9–16, February 2002.

[4] I. Albrecht, J. Haber, and H.-P. Seidel. Construction and Animation of Anatomically Based Human Hand Models. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 98–109. ACM SIGGRAPH, July 2003.

[5] J. Haber, K. Kähler, I. Albrecht, H. Yamauchi, and H.-P. Seidel. Face to Face: From Real Humans to Realistic Facial Animation. In *Proc. Israel-Korea Binational Conference on Geometrical Modeling and Computer Graphics*, pages 73–82, October 2001.

[6] J. Haber, N. Magnenat-Thalmann, D. Terzopoulos, T. Vetter, V. Blanz, and K. Kähler. *Facial Modeling and Animation – Eurographics 2002 Tutorial Notes*. Eurographics Association, September 2002.

[7] W.-K. Jeong, K. Kähler, J. Haber, and H.-P. Seidel. Automatic Generation of Subdivision Surface Head Models from Point Cloud Data. In *Proc. Graphics Interface 2002*, pages 181–188. A K Peters, May 2002.

[8] K. Kähler, J. Haber, and H.-P. Seidel. Dynamic Refinement of Deformable Triangle Meshes for Rendering. In *Proc. Computer Graphics International 2001*, pages 285–290, 2001.

[9] K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based Muscle Modeling for Facial Animation. In *Proc. Graphics Interface 2001*, pages 37–46, 2001.

[10] K. Kähler, J. Haber, and H.-P. Seidel. Dynamically refining animated triangle meshes for rendering. *The Visual Computer*, 19(5), pages 310–318, 2002.

[11] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: Generating animated head models with anatomical structure. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 55–64. ACM SIGGRAPH, July 2002.

[12] K. Kähler, J. Haber, and H.-P. Seidel. Reanimating the Dead: Reconstruction of Expressive Faces from Skull Data. *ACM Transactions on Graphics*, pages 554–561, July 2003.

[13] M. Tarini, H. Yamauchi, J. Haber, and H.-P. Seidel. Texturing Faces. In *Proc. Graphics Interface 2002*, pages 89–98. A K Peters, May 2002.

Awards

[1] J. Haber and K. Kähler. Medusa – A Facial Modeling and Animation System. SaarLB Science Award, 2001.