–Faces and Hands–

Modeling and Animating Anatomical and Photorealistic Models with Regard to the Communicative Competence of Virtual Humans

Dissertation

zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Irene Albrecht

Saarbrücken
2005

Datum des Kolloqiums: 22.12.2005

Dekan der Naturwissenschaftlich-Technischen Fakultät I:
Prof. Dr. Jörg Eschmeier

Mitglieder des Prüfungsausschusses:
Vorsitzender: Prof. Dr. Joachim Weickert
1. Gutachter: Prof. Dr. Hans-Peter Seidel
2. Gutachter: Prof. Dr. Volker Blanz
Akademischer Mitarbeiter: Dr. Bodo Rosenhahn

Abstract

In order to be believable, virtual human characters must be able to communicate in a human-like fashion realistically. This dissertation contributes to improving and automating several aspects of virtual conversations. We have proposed techniques to add non-verbal speech-related facial expressions to audiovisual speech, such as head nods for of emphasis. During conversation, humans experience shades of emotions much more frequently than the strong Ekmanian basic emotions. This prompted us to develop a method that interpolates between facial expressions of emotions to create new ones based on an emotion model. In the area of facial modeling, we have presented a system to generate plausible 3D face models from vague mental images. It makes use of a morphable model of faces and exploits correlations among facial features. The hands also play a major role in human communication. Since the basis for every realistic animation of gestures must be a convincing model of the hand, we devised a physics-based anatomical hand model, where a hybrid muscle model drives the animations. The model was used to visualize complex hand movement captured using multi-exposure photography.

The spoken word is only one modality that humans use in their interactions. It is complemented by voice quality, facial expressions, hand gestures, posture, touch, and several others. Most of the time, we use these channels without even noticing, and become aware of them only in case of inconsistencies. In fact, non-verbal cues play an astonishingly important role in human conversation. If verbal and non-verbal channels transmit conflicting information, adults, in contrast to children, rely more on the non-verbal than on the verbal part of the message. About 60% to 65% of the meaning in a social context (e.g. attitude, dominance / submissiveness) is conveyed non-verbally [BBW89]. In settings where imparting facts is the main purpose, more reliance is placed on words. Since people's decoding abilities disagree for the different modalities of communication, some people place more weight on the verbal statement, some on certain non-verbal channels, and some shift reliance between channels according to context.
The above considerations suggest that arbitrarily garnishing animations of speech with head and eye movement etc. is likely to cause bewilderment in users. To come across as convincing and believable, virtual characters must imitate human conversational signals as exactly as possible.
Although the problem has been tackled from all sides by computer science, there is still a long way to go. Due to the size of the area, we restricted ourselves to faces and hands.

We were fortunate to have access to an existing facial animation system based both on physics and anatomy, which is capable of realistic real-time animations. On this platform, we implemented different aspects of speech-related non-verbal communication in the face.
The first type of movement is mostly concerned with utterance and dialog organization. It underlines the structure of the utterance, highlights important segments, and helps to coordinate turn taking during conversation. A lot of this information is also present in the speech signal, from where it can be extracted and translated into appropriate facial expressions. Unlike in this top-down approach, text-to-speech systems generate similar information during their bottom-up linguistic analysis. We implemented modules for both scenarios.
Most facial animation systems are capable of displaying Ekman's six universal emotions joy, fear, anger, sadness, surprise and disgust. In ordinary human interaction, however, the majority of feelings are more differentiated and less intense. Based on an emotion model that combines dimensional representation and categorization, our algorithm predicts expressions of intermediate emotions of arbitrary intensity. The large number of possible shades of emotion suggests to derive new facial expressions of emotions from known ones. This work extends relevant research

by Tsapatsoulis et al. [TRK+02]. By integrating the facial animation system and an emotional text-to-speech system based on the same emotion model, we obtained a text-to-audiovisual speech system capable of transmitting feeling in the verbal, vocal, and facial channels.
Statistical approaches to face modeling achieve photorealism by learning geometrical and texture properties from examples. We built a facial modeling tool based on a morphable model of 3D faces. The model captures dependencies between facial features, enabling the system to automatically compute the most plausible completion for an underspecified face model. This makes it a useful tool to mold coherent 3D face models from vague mental images, as present, for example, in a designer's mind, or as described by an eyewitness to the police.
Models that reflect human anatomy and that are animated following physical laws facilitate the creation of plausible animations by restricting movements to their natural range and by exhibiting realistic deformations such as muscle bulges. Hands, through gestures, also play a major role in communication, making people sensitive to their movements. However, an animatable human hand model fit to hold a candle to the facial animation system did not exist. Deciding to fill the gap, we developed a physics-based hand model which reproduces human anatomy. Because the hand is even more complex than the face and we still aimed at interactivity, we introduced a hybrid muscle model. Pseudo muscles animate the movement of the bones by simulating the involved mechanics, and geometric muscles model muscle deformations. This hand model was used successfully in another project, where we used it to visualize a baseball pitcher's hand movement during ball release. Pose data for key frames was obtained using stroboscope photography. In the same way, we also captured decisive ball parameters. Visualizations of hand pose and ball flight together show cause and effect with respect to the ball trajectory and provide a valuable tool for analyzing an athlete's performance.

To sum up, the key contributions of this thesis are:

- techniques for animating non-verbal speech-related facial expressions. Depending on the input type, different methods are employed: for text input, intermediate processing results of an integrated text-to-speech system allow to deduce position and type of speech accompanying facial movement, while in the case of an input speech signal, the relevant information is extracted from the audio directly.
- an algorithm to generate facial expressions for a continuum of mixed emotions of arbitrary intensity. The method creates facial movement for mixed emotions using intensity scaling and interpolation between known facial expressions of the most similar emotions. Since both this algorithm and the coupled emotional text-to-speech system are based on the same emotion model, the system allows to generate consistent emotional audiovisual speech.
- a system to create coherent three-dimensional models of faces from vague recollections or incomplete descriptions as those given by an eyewitness to the police. The method builds on a morphable model of three-dimensional faces. Taking into account correlations among facial features, unspecified parts of the face are completed automatically to yield the most plausible face model given the user input.
- a physically based hand model designed after human anatomy. The hybrid muscle model consists of pseudo muscles and geometric muscles. Pseudo muscles animate the model by moving the bones according to mechanical laws, while geometric muscles deform the skin surface via a mass-spring system.
- a high-speed tracking system based on multi-exposure photography with which we captured the hand of a baseball pitcher during ball release as well as the trajectory and initial conditions of the flying ball.

Zusammenfassung

Um überzeugend zu wirken, müssen virtuelle Figuren auf dieselbe Art wie lebende Menschen kommunizieren können. Diese Dissertation hat das Ziel, verschiedene Aspekte virtueller Unterhaltungen zu verbessern und zu automatisieren. Wir führten eine Technik ein, die es erlaubt, audiovisuelle Sprache durch nichtverbale sprachbezogene Gesichtsausdrücke zu bereichern, wie z.B. Kopfnicken zur Betonung. Während einer Unterhaltung empfinden Menschen weitaus öfter Emotionsnuancen als die ausgeprägten Ekman'schen Basisemotionen. Dies bewog uns, eine Methode zu entwickeln, die Gesichtsausdrücke für neue Emotionen erzeugt, indem sie, ausgehend von einem Emotionsmodell, zwischen bereits bekannten Gesichtsausdrücken interpoliert. Auf dem Gebiet der Gesichtsmodellierung stellten wir ein System vor, um plausible 3D-Gesichtsmodelle aus vagen geistigen Bildern zu erzeugen. Dieses System basiert auf einem Morphable Model von Gesichtern und nutzt Korrelationen zwischen Gesichtszügen aus. Auch die Hände spielen ein große Rolle in der menschlichen Kommunikation. Da der Ausgangspunkt für jede realistische Animation von Gestik ein überzeugendes Handmodell sein muß, entwikkelten wir ein physikbasiertes anatomisches Handmodell, bei dem ein hybrides Muskelmodell die Animationen antreibt. Das Modell wurde verwendet, um komplexe Handbewegungen zu visualisieren, die aus mehrfach belichteten Photographien extrahiert worden waren.

Das gesprochene Wort ist nur eine der Modalitäten, die Menschen während einer Unterhaltung nutzen. Es wird durch Stimmqualität, Gesichtsausdruck, Gestik, Körperhaltung, Berührung u.v.m. ergänzt. Meistens benutzen wir diese Kanäle, ohne es zu bemerken, und werden uns ihrer nur im Falle von Unstimmigkeiten bewußt. Tatsächlich spielt Körpersprache eine erstaunlich große Rolle in der zwischenmenschlichen Kommunikation. Wenn verbale und nichtverbale Kanäle widersprüchliche Informationen übermitteln, verlassen sich Erwachsene – anders als Kinder – eher auf den nichtverbalen als auf den verbalen Teil der Nachricht. Etwa 60% bis 65% der Bedeutung in einem sozialen Kontext (z.B. innere Einstellung, Dominanz / Unterordnung) werden durch die Körpersprache ausgedrückt [BBW89]. In Situationen, in denen es hauptsächlich um die Übermittlung von Fakten geht, wird den Worten größere Bedeutung beigemessen. Da sich die Dekodierungsfähigkeiten einzelner Personen für die verschiedenen Kommunikationskanäle unterscheiden, gewichten manche Leute die verbale Aussage stärker, andere die nichtverbale, und wieder andere verteilen ihr Vertrauen je nach Kontext auf die verschiedenen Kanäle.
Die obigen Überlegungen legen nahe, daß die Benutzer vermutlich mit Befremden reagieren würden, wenn man Sprachanimationen mit beliebigen Kopf- oder Augenbewegungen etc. unterlegte. Um überzeugend und glaubwürdig zu erscheinen, müssen virtuelle Figuren menschliche Kommunikationssignale so genau wie möglich imitieren.
Obwohl das Problem in vielen Bereichen der Informatik untersucht wird, liegt eine umfassende Lösung noch in weiter Ferne. Da dieses Gebiet sehr umfangreich ist, beschränkten wir uns auf Gesichter und Hände.

Vorteilhafterweise konnten wir auf ein existierendes Gesichtsanimationssystem zurückgreifen, das sowohl physik- als auch auf anatomiebasiert ist und realistische Animationen in Echtzeit erzeugt. Auf diesem Fundament implementierten wir verschieden Aspekte von sprachbezogener nichtverbaler Kommunikation im Gesicht.
Die erste Art von Bewegung dient hauptsächlich der Organisation von Äußerungen und Dialogen. Sie unterstreicht die Struktur einer Äußerung, hebt wichtige Teile hervor und hilft bei der

wechselseitigen Dialogkoordination. Einen Großteil dieser Informationen kann man auch aus dem Sprachsignal ableiten und in entsprechende Gesichtsausdrücke übersetzen. Anders als in diesem Ansatz, wo man von real Gegebenem ausgeht, generieren spracherzeugende Systeme ähnliche Informationen während der linguistischen Analyse des Eingabetextes. Wir implementierten Module für beide Szenarien.
Die meisten Gesichtsanimationssysteme können Ekmans sechs allgemeingültige Emotionen Freude, Angst, Wut, Traurigkeit, Überraschung und Ekel darstellen. In menschlichen Unterhaltungen ist die Mehrheit der Gefühle gewöhnlich aber differenzierter und weniger stark ausgeprägt. Ausgehend von einem Emotionsmodell, das dimensionale Darstellung und Kategorisierung verbindet, sagt unser Algorithmus Gesichtsausdrücke von emotionalen Zwischenzuständen beliebiger Intensität voraus. Die große Anzahl der möglichen Emotionsschattierungen legt es nahe, neue emotionale Gesichtsausdrücke aus bereits bekannten abzuleiten. Diese Arbeit entwickelte einen entsprechenden Ansatz von Tsapatsoulis et al. [TRK$^+$02] weiter. Die Integration des Gesichtsanimationssystems und eines Systems, das emotionale Sprache ausgehend von demselben Emotionsmodell erzeugt, versetzte uns in die Lage, ein Gefühl mit Worten, Stimme und Gesicht auszudrücken.
Statistische Gesichtsmodellierungsansätze erreichen Photorealismus, indem sie Geometrie- und Textureigenschaften aus Beispielen lernen. Wir bauten ein Programm zur Gesichtsmodellierung auf, das auf einem Morphable Model von 3D-Gesichtern basiert. Das Modell erfaßt Abhängigkeiten zwischen Gesichtszügen, sodaß das System automatisch die plausibelste Ergänzung eines unterbestimmten Gesichts berechnen kann. Dies macht es zu einem nützlichen Werkzeug, um stimmige 3D-Gesichtsmodelle aus vagen geistigen Bildern herauszuarbeiten, wie sie z.B. einem Designer vorschweben oder der Polizei von Augenzeugen beschrieben werden.
Modelle, die die menschliche Anatomie widerspiegeln und physikalischen Gesetzen folgend animiert werden, erleichtern es, plausible Animationen zu erzeugen, indem sie Bewegungen auf ihren natürlichen Bereich einschränken und realistische Verformungen, wie z.B. Muskelschwellungen, aufweisen. Da in der gestischen Kommunikation auch die Hände eine große Rolle spielen, beobachten Menschen Handbewegungen sehr genau. Ein animierbares menschliches Handmodell, das dem Gesichtsanimationssystem das Wasser reichen konnte, gab es nicht. Um diese Lücke zu schließen, entwickelten wir ein physikbasiertes Handmodell, das die menschliche Anatomie wiedergibt. Da die Hand sogar noch komplexer als das Gesicht ist, wir aber Interaktivität anstrebten, führten wir ein hybrides Muskelmodell ein. Pseudomuskeln animieren die Bewegung der Knochen, indem sie die relevante Mechanik simulieren, und geometrische Muskeln modellieren Muskelverformungen. Dieses Handmodell wurde erfolgreich in einem weiteren Projekt dazu eingesetzt, die Handbewegungen eines Baseballwerfers im Moment des Abwurfs zu visualisieren. Die Handhaltung an den Keyframes wurde durch Stroboskopphotographie ermittelt. Auf diese Art maßen wir auch wichtige Ballparameter. Visualisierungen von Handbewegung und Ballflug zusammen zeigen Ursache und Wirkung in Bezug auf die Flugbahn des Balls und bieten ein wertvolles Werkzeug, um sportliche Leistung zu messen.

Zusammenfassend sind die wichtigsten Beiträge dieser Arbeit:

- Ein Verfahren zur Animation von nichtverbalen sprachbezogenen Gesichtsausdrücken. Abhängig von der Art der Eingabe werden verschiedene Methoden eingesetzt: Für Texteingaben erlauben Zwischenergebnisse eines eingebundenen spracherzeugenden Systems, Position und Art der sprachbegleitenden Gesichtsbewegungen abzuleiten, während im Fall eines Eingabesprachsignals die relevanten Informationen aus dem Ton gewonnen werden.

– Ein Algorithmus, um Gesichtsausdrücke für ein Kontinuum von gemischten Emotionen von beliebiger Intensität abzuleiten. Die Methode erzeugt Gesichtsbewegungen für gemischte Emotionen durch Skalierung der Intensität und Interpolation zwischen bekannten Gesichtsausdrücken der ähnlichsten Emotionen. Da sowohl dieser Algorithmus als auch das angeschlossene spracherzeugende System auf demselben Emotionsmodell beruhen, erlaubt das System, konsistente emotionale audiovisuelle Sprache zu erzeugen.
– Ein System, um stimmige 3D-Modelle von Gesichtern aus vagen Erinnerungen und unvollständigen Beschreibungen zu erzeugen, wie sie die Polizei von Augenzeugen erhält. Dieses Verfahren baut auf einem Morphable Model von dreidimensionalen Gesichtern auf. Indem Korrelationen zwischen Gesichtszügen berücksichtigt werden, werden unspezifizierte Teile des Gesichts automatisch ausgefüllt, um das plausibelste Gesicht für die Benutzereingabe zu erhalten.
– Ein auf Physik basierendes Handmodell, das der menschlichen Anatomie nachempfunden ist. Das hybride Muskelmodell besteht aus Pseudomuskeln und geometrischen Muskeln. Pseudomuskeln animieren das Modell, indem sie die Knochen nach mechanischen Gesetzen bewegen, während geometrische Muskeln die Hautoberfläche mit Hilfe eines Masse-Feder-Netzwerkes verformen.
– Ein Hochgeschwindigkeitssystem zur Bewegungserfassung, das auf Mehrfachbelichtung von Photographien aufbaut. Mit diesem Werkzeug fingen wir sowohl die Handbewegungen eines Baseballwerfers zum Zeitpunkt des Abwurfs ein, als auch die Wurfbahn und die Anfangsbedingungen des fliegenden Balls.

Acknowledgements. First and foremost, I am grateful to Hans-Peter Seidel for rendering this thesis possible and for making his group what it is. Jörg Haber was a great and knowledgeable advisor, and more than just that. My colleagues made the MPI a great place to be. Working together with Jörg Haber, Kolja Kähler, Christian Theobalt and Volker Blanz was not only very instructive and constructive, but – better still – a lot of fun. Jacques Koreman from the institute for computational linguistics and phonetics of the Universität des Saarlandes greatly supported our work on speech-synchronized facial animation by giving invaluable advice and background information and by selflessly performing many phonetical transcriptions. Furthermore, I would like to thank Marc Schröder from the DFKI in Saarbrücken for fertile collaboration and for the contagious ardor with which he approached our projects. Special thank is due to Thorsten Dehm from the Saarlouis Hornets for enthusiastically performing dozens of pitches for the baseball project, and to Herrn Hans from the LKA Saarland for sharing his experience as a forensic artist and for participating in our experiments on facial composite creation. Finally, I am most grateful to my family and Stefan Burkhardt for their unvarying and patient encouragement and support.

Contents

1

Introduction

One cannot not *communicate.*
– Paul Watzlawick

The famous first axiom of Paul Watzlawick's communication theory may sound a bit scary, but it is undoubtedly true: even utter immobility, even complete nonsense makes a statement. The most natural means of communication for any creature is its body. Within the human body, the channel capable of the most complex communication is the voice, followed by the face and the hands.
Complicated ideas are best expressed by words. But words are not the only carriers of information in the voice channel: the manner of speaking conveys a lot of information as well. From variables such as pitch, loudness, and pauses one can identify new or otherwise important parts of the utterance as well as deduce the speaker's currently felt emotion.
Direct visible manifestations of speech such as lip and tongue movement enhance speech understanding, even for people that are not hearing impaired. The opposite, i.e. how annoying asynchronous speech and lip movement can be, is obvious, for instance, from badly dubbed movies. In its most unfortunate form, inconsistency between lip movement and sound can even inhibit correct understanding [MM76].
Other facial expressions linked to speech include eyebrow and head movement, blinking, nose wrinkling, and the like. Mostly, they serve to distinguish prominent parts of speech, to structure the utterance, and to regulate turn taking. Unless consciously suppressed or masked, emotions are strongly visible in the face as well.
After voice and face, the hands are doubtless the part of the body most often and consciously employed by humans during communication. "Speaking with one's hands" is also a source of many jokes, both because of the cultural differences in frequency of hand movement and because of the differences in meaning of individual signals. All channels of communication depend on culture, not only language. This extends to every aspect of communication, including non-verbal speech-related signals of every kind, and to some extent even to expressions of emotion.
Other means of communication include posture, which tells a lot about a person's current frame of mind and about his relationship to his communication partner, distance, i.e. how close together or how far apart the participants in the conversation are, touching behavior, timing, clothing, and artifacts such as jewelry.

Its ubiquity, its interdisciplinary character, and the plethora of possible applications make human communication a challenging and rewarding field of research that is of interest not only to communication theorists, linguists, psychologists, doctors, politicians, teachers and any other person from public life, but increasingly so to computer scientists as well. Since computers form part

of an ever increasing number of aspects of our daily lives, realistic simulation of human communication is important for human computer interfaces, virtual sales persons, electronic kiosks, for avatars in chatrooms, and for characters in computer games. In addition, movie actors are becoming more and more computerized. Another interesting aspect is the feedback that is provided with regard to communication models.
Several directions have developed in computer science that investigate different aspects of communication. Artificial intelligence is interested in dialog management for virtual agents, including personality traits and emotions. Text to speech systems are constantly improving and are now even able to produce emotional speech. Computer graphics strives to implement the visible aspects of communication through human modeling and animation, while computer vision handles tracking of eyes, hands and the entire body. Virtual believable agents, finally, bring it all together.

Our contributions lie in the area of computer graphics, more precisely in the rule-based modeling of non-verbal, speech-related facial expressions, in the generation of non-basic facial expressions of emotion, in the development of a three-dimensional facial composite system capable of automatically completing underspecified face models in a plausible way, in the design of an animatable human hand model, and, veering towards computer vision, in devising a high-speed tracking system based on multi-exposure photography.
This thesis is organized as follows: first, we will put the work presented here in perspective by giving an overview of related work (Chapter 2). In Chapter 3, the two face models on which much of our work builds are described, The subsequent chapter details our contributions in the area of non-verbal facial animation, both directly speech-related (Sections 4.2, 4.3) and with regard to emotions (Section 4.4). Chapter 5 explains the facial composite system, and Chapter 6 finally deals with our approach to hand modeling, hand animation, and high-speed tracking of hand and ball during baseball pitches. A general discussion concludes the thesis.

2

Related Work

This chapter summarizes work that is related to ours. Starting from a brief introduction to research on facial animation in general (Section 2.1) and lip sync in particular (Section 2.1.1), the first section will proceed to non-verbal speech-related facial animation (Section 2.1.2) and to facial expressions of emotion (Section 2.1.3). Section 2.2 deals with facial composite systems. It presents research work, systems used in law enforcement, and programs for custom tailoring virtual characters for computer games. Section 2.3 gives an introduction to research on the human hand, both in the area of biomechanics and related disciplines, and in computer graphics. The remainder of the chapter (Section 2.4) reports on literature relevant to tracking and modeling baseball pitches.

2.1 Facial Animation

Facial animation has been a field of active research since the 1970's. Good overviews of the area can be found in [PW96, NN99, BBEO03]. Apart from keyframe interpolation, the main techniques used in facial animation are performance driven approaches and direct parameterizations as well as pseudo muscle- and muscle-based techniques. Often a combination of these approaches is used.

The *performance-based* approach [Wil90] uses data of real human motion to drive the animation. Motion is captured by tracking feature points, like the corners of the mouth, the tip of the nose etc., or markers attached to the face [Wil90, GGW$^+$98, CLK01]. In [GGW$^+$98], 3D animations are generated from video. Simultaneously with the marker tracking, video images are captured as texture maps for a three-dimensional face model obtained with a laser scanner. Choe et al. [CLK01] propose a combination between a performance driven approach and a pseudo muscle-based approach. Since performance-based systems use actual human movement data to drive the animations, they yield the most natural animations. Transferring the data to new individuals can be a problem, though, especially in the case of motion capture data or of video models.

Example-based techniques learn facial movement from video [BCS97, CG98, Bra99, EGP02, CE05, CTFP05]. Wang et al. [WHL$^+$04] use a high-speed 3D scanner to capture moving faces. They learn generic expressions and personal expression styles from the data. Blanz et al. [BBPV03] fit a morphable model of three-dimensional faces (see Section 3.2) to a face in an image or video. They change the expression of the obtained 3D model by adding learned expression vectors. The modified face is rendered back into the original image with the appropriate pose and illumination parameters. Recently, Vlasic et al. [VBPP05] developed

a system based on a multilinear model that captures head pose, facial expression and viseme from 3D scans. When the model is fit to the frames of a video, it allows captured motion to be transferred to a 3D face model of a different person.

In *direct parameterized* models [Par74], vertices of a polygon or spline face model are grouped together and assigned a joint parameter. The nodes belonging to one parameter are displaced directly according to the parameter value. Examples for this approach include the MPEG-4 facial animation parameters (FAPs) [PF02] and [Par82, PWWH86, MTPT88, KMMTT92, KP05]. This is the most flexible method, since it allows unconstrained displacement of vertex groups. If realistic animations are desired, the responsibility to restrict animations to natural movements lies with the animator.

Pseudo muscle-based models [PB81, Wat87, CLK01] simulate muscle movement using geometric deformation operators. Muscles are modeled as forces which deform the facial geometry, but do not have a volume. Both mass spring networks [PB81, Wat87] and finite elements [CLK01] lend themselves to building a physics-based skin model on which the muscle forces can act.

In contrast to pseudo muscle-based models, *muscle-based* approaches [TW90, WF95, LTW95, KHS01, Käh03, Gla03, SNF05] assign a geometric shape to every muscle. Since skin and connective tissue are modeled either by mass spring systems [TW90, WF95, LTW95, KHS01, Käh03] or by finite element models [Gla03, SNF05], muscle deformations are propagated to the skin. As a consequence, these are the physically and anatomically most exact models, but they are also computationally expensive. In a combination of the muscle-based and performance-based approach, Sifakis et al. [SNF05] determine muscle activation values from motion capture marker data. The facial animation system by Kähler et al. [KHS01, Käh03] is treated in more detail in Section 3.1.

Facial animation is a wide area of research. It is not only concerned with the different animation techniques, but also with how to generate convincing facial animation as automatically as possible. It comprises animation of lip movement during speech (see Section 2.1.1), non-verbal speech-related movements such as lifting the eyebrows on accented syllables (Section 2.1.2), and facial expressions of emotion (Section 2.1.3). Mostly, every category has been treated in isolation. This raises the question of how to combine the different kinds of movement. Usually, this is done in an additive fashion (e.g. [CPB+94, PBS96, KP05]) without giving a lot of thought to it. Bui et al. [BHN04] explicitly address the problem of integrating different channels of facial expression, i.e. lip sync, conversational displays, emotional expressions, etc., for muscle-based models. Muscle contractions from different channels are combined according to priority, taking into account conflicts at the muscle level. Pelachaud and Poggi [PP02] use Bayesian networks to resolve conflicts between facial expressions resulting from co-occurrence of communicative functions. In [CDB02], Chuang et al. propose to split motion capture data into content and style, i.e. into visemes and concurrent emotional expressions. Animations can then be modified to display the same content with a different emotion display. Both analysis and synthesis are achieved using a factorization model. Another learning based approach is the one by Cao et al. [CTFP05]. It decomposes the captured motion into speech and emotion using independent component analysis [CFP03]. The method derives a mapping between discrete emotion spaces from the training data, which permits the modification of emotions in speech synchronized animations. Emotions for new lip-synched animations are either specified by the animator or deduced from the speech

signal by a support vector machine. Based on a multilinear model of 3D face scans, Vlasic et al. [VBPP05] factorize video into visemes, expressions, and identity. The motion data can be edited, combined, and transferred to other identities within the model.

2.1.1 Lip Sync

Since this thesis centers on communication, touching the work on facial animation of mouth movement for speech at least briefly is mandatory. For the sake of brevity, only approaches which allow for coarticulation are mentioned. The term *coarticulation* refers to the influence of surrounding phonemes on the vocal tract shape of a segment, and hence on the lip shape of the segment. Lip sync that does not take this effect into account yields animations of perceivably lower quality.

The rule-based approach to by Pelachaud et al. [PBS91, PBS96] explicitly states the susceptibility of individual phonemes to coarticulation. It considers also the physical properties of the vocal tract, which impose certain timing constraints on facial movements. Other systems [CM93, LG97, CG98, DRSV02, KP05] assign dominance functions to every phoneme in order to model the influence of a segment on the surrounding visemes[1] (cf. Section 3.1.3). Another technique is to concatenate prerecorded polysemes[2] from a database [BCS97, CG98]. Cao et al. [CFKP04] identify the sequence of video chunks that best matches an input phoneme string using a greedy graph search algorithm and then stitch the pieces together to obtain the final animation. Learning-based approaches [BS98b, GUAT98, EGP02, KMvG04] learn facial movement and hence coarticulation from video footage. To a certain extent, it is also possible to extract information on coarticulation effects from the attributes of the acoustic speech signal [KMT00]. Waters and Levergood [WL93] construct a mass spring network from the mesh nodes around the mouth. Solving the equations of motion for the animations will approximate target mouth shapes instead of interpolating them, thereby leading to some kind of coarticulation.

2.1.2 Non-verbal Speech-related Facial Animation

In recent years, facial animation systems have reached a degree of realism that allows creation of photo-realistic full-feature movies. However, the animation process is still enormously time-consuming, especially for speech-synchronized facial animations, which to this day are mostly hand crafted. A fully automatic method to generate facial animation from audio or simple text is thus a much desired goal. Apart from lip sync (see Section 2.1.1), a major problem is that huge background knowledge is necessary to correctly interpret the meaning of written sentences, and to transfer this meaning to appropriate facial expressions. A question might be a rhetorical one, a remark can have an ironic touch. These subtleties should be reflected in the face of the speaker. Similarly, in the case of audio input, this information would ideally be extracted from the speech signal. A lot of interesting research has been carried out in this area. The following paragraph introduces relevant psychological and paralinguistic foundations. It is succeeded by a passage on results from computer graphics.

[1] A *viseme* is the visual counterpart of a phoneme, i.e. the minimal visual building block of speech.

[2] A *polyseme* comprises several visemes.

Psychologic and Paralinguistic Research

The information that is not contained in the words themselves, but in the "acoustic packaging" of the utterance [Bar01, p. 597], e.g. in prosody or frequency and duration of pauses, is referred to as *paralinguistic* information. The conjunction of paralinguistics and psychology is able to describe the correlation between prosody and facial expressions. A lot of valuable information for speech-synchronized non-verbal facial animation can be drawn from this interdisciplinary field of research.

The relation of speech and eyebrow movement was systematically investigated by Ekman [Ekm79]. His pioneering research indicates that certain words and also greater parts of a sentence are often accompanied by raising or lowering of both the inner and the outer part of the brows. He called these facial gestures *batons*, when only one word is emphasized, or *underliners* for multiple words. The type of movement depends largely on context: the brows will most probably be lowered in situations of perplexity, doubt or other difficulties. Eyebrow movements also serve as *punctuators*, i.e. they are used similarly to punctuation marks in written text. Again, lowered brows indicate difficulty, doubt, or perplexity, but also seriousness and importance. To show that a question is being asked, eyebrows are often raised. During pauses caused by the speaker's searching for words, raised brows occur accompanied by an upward gaze direction. Looking at a still object to reduce visual input is another typical behavior for word searches. Especially in conjunction with an '*errr*' sound, eyebrows may also be lowered in this situation.
Chovil [Cho91] reports that *syntactic displays* (batons, underliners, punctuators, etc.) are the most frequent speech accompanying facial gestures. Among these, raising or lowering of brows are most prevalent. Other important movements of the speaker are related to the content of the speech, e.g. facial shrugs or expressions while trying to remember something.
Cavé et al. [CGB+96] investigated the link between eyebrow movement and pitch contour. In 71 % of the examined cases a correspondence was found, where rise and fall of the pitch of the speech signal coincided with raising and lowering of the speaker's eyebrows, respectively. Typically, 38 % of overall eyebrow movements occur during pauses or while listening. They serve to indicate turn taking in dialogs, assure the speaker of the listener's attention, and mirror the listener's degree of understanding, serving as back-channel. House et al. [HBG01] examined the importance of eyebrow and head movement for the perception of significance. They observed that both movements are weighty here. Perceptual sensitivity to timing is around 100 ms to 200 ms, which is about the average length of a syllable. Investigating the relationship between questions and gestures, Cosnier [Cos91] found that for informative questions (i.e. not related to the interaction itself) head and eyebrow movements do not differ from normal informative conversation, with the exception of raising the head and possibly the eyebrows at the end of a question. However, the visual focus is more often on the listener than during statements. Relations between emotions (joy, fear, anger, disgust, sadness, boredom) and prosodic parameters (**F0** floor/range/slope, jitter, spectral energy distribution, number of accentuated syllables) have been investigated, for example, by Paeschke et al. [PKS99] and by Johnstone and Scherer [JS99]. They report that most of the measured prosodic parameters are suitable to classify emotions.

Computer Science

The integration of synchronized speech animation with facial expressions has been carried out by Pearce et al. [PWWH86] and by Ip and Chan [IC96], both using a script-based approach:

the expression to be displayed during speech is specified by the user in a domain-specific script language. Kalra et al. [KMMTT91] describe a layered, script-based approach to specify facial animations. Similar to theses approaches, the RUTH system by DeCarlo et al. [DRSV02] takes as input text annotated with facial expressions. Audible speech is generated by a text-to-speech system, which also returns timing information for the non-verbal facial expressions and a timed phoneme string. Based on this phoneme representation, speech-synchronized mouth movements are computed similarly to Cohen and Massaro [CM93].
In contrast to explicit scripting techniques, the image-based system proposed by Brand [Bra99] learns the dynamics of real human faces during speech using original video footage. This information is then applied to create speech animations from novel audio input. The system generates mouth movements including coarticulation as well as additional speech-related facial animation, for instance eyebrow movement. Lee et al. [LBB02] developed a model for human ocular behavior during communication, based on empirical models of saccades and statistical models of eye tracking data.
Text-to-speech techniques have been used by Pelachaud et al. [PBS96] and by Cassell et al. [CPB+94] for synthesis of speech-synchronized animations of agents interacting with each other or with the user. The component that generates the text for the agent's speech has additional knowledge about content and structure of a piece of dialog, which is employed to generate appropriate gestures. In their more recent work, Poggi and Pelachaud [PP00, PP02] include the dialogue situation into their animations, i.e. they distinguish between semantics and performative act. For example, you will probably make suggestions to your boss with a different attitude than when you order your children to do something, although the actual content of your utterance may be more or less the same. Lundeberg and Beskow [LB99] have developed a spoken dialog system featuring a virtual representation of the famous Swedish writer Strindberg. Similar to [PBS91, CPB+94], the agent is capable of communicating using bimodal speech augmented by simple punctuation gestures like nods or blinks. More complicated gestures have been explicitly designed for certain characteristic sentences. Scott et al. [KKM03] combined a bi-lingual dialogue manager and a talking head. In addition to the text of the utterance, the dialogue manager provides instructions for different non-verbal behavior for English and Maori, respectively.

2.1.3 Facial Expressions of Emotion

The face is also a very important channel for communicating emotions. Ekman [EK97] identified a set of six basic emotional facial expressions that are valid throughout all cultures: joy, anger, fear, disgust, sadness, and surprise. Many facial animation systems can display these universal expressions of emotion. However, the human face is capable of many more emotional expressions, but little research has been conducted in this direction so far, mainly due to the limited availability of data. Since our contribution consists in generating the more subtle expressions of non-basic emotions (see Section 4.4), the following paragraphs will summarize previous work in facial animation concerned with expressing mixed feelings.

The FacEMOTE system [BB02] relies on the Laban Movement Analysis of body motion which has been transferred to the face. The method modifies an input facial animation stream to change its expressiveness. The four parameter pairs used to steer the process are direct-indirect, light-strong, sustained-quick, and free-bound. A direct mapping from these parameters to emotions is not provided. Bui et al. [BHPN01] propose a fuzzy rule based system to map emotions to muscle contraction values. The system comprises a set of rules both for the display of single emotions,

and for two simultaneous emotions. In the latter case, the facial expressions of the two emotions are restricted to non-overlapping regions of the face to avoid conflicts. Ruttkay et al. [RNtH03] arrange the six basic emotions equidistantly on the border of a disc according to similarity. To every point on the disc a facial expression is associated which is computed by linear interpolation between the closest basic emotions. Distance from the circle center describes intensity. In the same paper, the authors present a second method to obtain new expressions based on principal component analysis (PCA). However, they did not find the significant principal components to be an intuitive means for searching the space of emotional facial expressions. Tsapatsoulis et al. [TRK+02] (Section 4.4.3) have also developed a method to interpolate between affect displays to create new ones. They use a mixture of two emotion models, Whissell's activation-evaluation approach [Whi89] (cf. Section 4.4.1) and Plutchik's emotion wheel [Plu80]. Latta et al. [LAAB02] make use of the activation-evaluation coordinate system as well, but only for navigation through a predefined space of facial expressions and not for expression generation.

2.2 Facial Composite Systems

Facial composites are an important tool in law enforcement. Although many different commercial systems are in use, there is still room for improvement. This has lead to a number of research projects in computer science. Followed by an overview of commercial systems, they are considered below. A similar problem is posed by increased demands on character customizability in computer games.

2.2.1 Computer Science

Approaches to composite creation in research that do not follow in the footsteps of the classical Identi-Kit toolkit [Smi05] often rely on modifications of the coefficients obtained from PCA, or a combination of the two techniques. It has been demonstrated that PCA is very well suited to describe face space (e.g. [BV99]), but the problem with this approach lies in the unintuitiveness of the individual principal components. They simply do not describe cognitively useful categories. Therefore a main issue when incorporating this technique into a facial composite system must be the design of a wrapper to ensure comfortable operation. The focus of the paper by Chen and Fels [CF04] lies on the appropriateness of different user interfaces for navigating face space with the main principal components of the example faces as coordinate axes.
In computer science, the motivation for developing a facial composite system was often research in the field of database retrieval, or more specifically, of finding suspects in a mug-shot data base [WAL+94, BM96, BS98a]. The PCA coefficients of the composites are used as access keys in the search. Brunelli and Mich [BM96] apply PCA to the individual features of the faces in a mug-shot database. The composite face used to search the database is constructed starting from the average of all faces in the data base. Image modification is achieved through changing the PCA coefficients of individual features directly, by selecting feature coefficients by keyword, or by importing face parts from the database of mug shots. During composite creation, the system displays and automatically updates the faces from the database that are most similar to the current composite. Baker and Seltzer [BS98a] use PCA on the entire face to determine similarity for mug-shot database search. Composite images are obtained by cutting and pasting features from images in the database, or by random combination of features of user-selected faces. The facial composite module of the database retrieval system by Wu et al. [WAL+94] works similar to the classical Identi-Kit tool, i.e. faces are composed from individual features

in a database. The similarity measure of faces is based on landmarks and on PCA of individual features. The system addresses the problem of aging.
Several approaches make use of the fact that the human brain is better equipped to recognize faces than to describe them. They employ genetic algorithms to approximate the target face. The user is shown a selection of faces, from which he chooses the most similar ones. These are then interbred and mutated. The process is repeated until the desired face crystallizes. The first system of this kind was the one by Caldwell and Johnston [CJ91]. They consider five different facial features, and assemble composites from an example database. Genes are defined as a concatenation of code for example type and position of the individual features. A descriptive language for faces is proposed in [DiP02]. It is parameter-based and can also be used to specify animations. A genetic algorithm is employed for navigation through face space. The system was designed to create 3D heads for the computer game "The Sims". The 2D facial composite system in [GPBS03] works by running an evolutionary algorithm on a facial appearance model. To obtain an appearance model, one has to perform an independent PCA both for texture and shape, and another PCA to combine the two into the final model. In order to allow for modifications at the local level, analogous appearance models for individual features were built. Frowd et al. [FHC04] compute individual PCAs on the shape and on the texture of the training images. New faces are evolved with a genetic algorithm where the PCA coefficients serve as genes. Shape and texture are treated separately. Shifting features is also possible.
To our knowledge, the only other composite system to take into account statistical correlations between facial features is the one by Gillenson and Chandrasekaran from 1975 [GC75]. It produces line drawings of a face. A statistical average face is used as starting point, and then modified by applying affine transformations and intensity changes to individual features or by importing features from a database. The user is prompted to deal with features in the order of importance for recognition. Statistical feature correlation is only considered in the beginning, when the overall shape of the face is established through stretching. Aging is simulated by introducing wrinkle lines.

2.2.2 Commercial Systems

Facial composite systems in use by law enforcement agencies all over the world work more or less in the same manner: the witness assembles the target face from parts of faces in a database. Most of the systems are grayscale and only support front views. They are basically software implementations of the classical Identi-Kit system [Smi05]. Laughery and Fowler [LF80] found that artist sketches are superior to Identi-Kit images due to the limited amount of facial features and lack of shading in the Identi-Kit toolkit. Both points of criticism have improved with today's computer-assisted composite creation, since database size has increased substantially. With most systems, shading is now integrated in the example features, especially in those instances where face parts from photographs are used.
PROfit™ [ABM05] is a traditional facial composite system where a face is composed from individual features. The feature database can be extended using drawing software. A database with features in 3/4 view is also available to either directly construct composites from this perspective or to automatically generate 3/4 views from frontal composites. Composite faces can be aged or caricatured. Deffenbacher et al. [DJVO00] demonstrated that caricaturing a face facilitates recognition. With PROfit, morphing between composites from different witnesses is possible to improve identification. It has been shown [BNH$^+$02] that a combination of the mental images of several witnesses is at least as good as the best individual likeness. E-FIT™ [Asp05] allows

the user to transform features or to select them from a database. In order to avoid confusing the witness by showing him too many faces dissimilar to his mental image, or even disembodied features, the features in the database are rendered into the current composite. Where the witness cannot give a description, default features are available. An extension is offered where a 3D mesh is fitted to and textured with the composite image. Aging of faces is possible, as well as importing a background image to recreate a crime scene. With the color photofit system PHANTOM PROFESSIONALxp© [UNI05], the source selects one starting image that comes closest to the target face in overall face shape, skin color, age etc. This base image is modified by affine transformations and by importing features from other faces in the database. Aging by approximately ten years is supported as well. If the faces in the data base are also available as side-views, a profile can be generated together with the front view. Transformations and retouching, however, must be done separately for each view. Starting from the witness's description, the FACETTE® composite system [IDE05] generates a number of suggestions, from which the source selects the most appropriate one. This face is then improved by importing features from a data base of face parts. The components can be adjusted with regard to brightness and contrast. Smith&Wesson® also developed facial composite software, Identi-Kit.NET™ [Smi05]. As the name suggests, the basic principle is very much the same as with the original Identi-Kit tool. In order to allow better coordination between law enforcement agencies, composites can be published in an online database. Aging software is offered as an add-on. FACES [IQ 05] is another simple black and white facial composite system, where composites are assembled from facial features in a database. Aging is simulated by overlaying wrinkle lines over the completed composite. An identification code is generated for every composite from the code for the individual features to allow low bandwidth transmission and reassembly at the other end.

2.2.3 Computer Games

Generating faces from mental images is also of interest for the entertainment industry. Several games include tools that allow the player to custom-tailor his character. Examples are *FIFA Soccer 2005*, *The Sims 2*, or *EVE Online*, to name but a few. These systems modify faces by morphing. Depending on the number of parameters, this can give the user many degrees of freedom, but importing features from a database is not possible. Non-natural faces can be generated by choosing certain parameter combinations. This is desirable with some game types, but certainly not with all. Naturally, since the faces are meant for hardware rendering, the underlying meshes cannot be very fine. Detail is added through textures.

2.3 Hand Models

The role of hands in communication is only surpassed by that of the face. In computer graphics, human hands are only now starting to receive the attention they deserve. Disciplines related to medical science have a decided head start here – and fortunately so, since results from anatomy, biomechanics and anthropometry must form the basis of all attempts at realistically modeling this sophisticated body part.

2.3.1 Anatomy, Biomechanics, and Anthropometry

Research in anatomy and biomechanics has shown that the human hand is a very intricate and elegant mechanical device, where many dedicated parts cooperate in an highly optimized inter-

play to form a powerful whole. Information on the anatomical building blocks of the hand can be found in illustrated anatomy books [PP01] or in more detail in [Cha90]. The book by Brand and Hollister [BH99] is inclined more towards biomechanics: meant as a textbook for hand surgeons, for instance when planning a tendon transfer operation, it provides a thorough description of the functioning of the hand.

Landsmeer [Lan61] developed a physics-based model for determining tendon excursion from joint angle, depending on the way the tendon crosses the joint. Starting from this model, he derived criteria of how muscles must be arranged in a joint system to be able to move the joints in any given way. In [AUC+83], tendon excursions of the index finger muscles have been measured and the corresponding moment arms have been computed using Landsmeer's tendon models.

A kinematic model for flexion and extension of the fingers has been developed by Lee and Kroemer [LK93]. Their model is based on the assumption that the moment arms of the tendons at the joints are constant. Considering external forces affecting the joints, they compute the finger strength for the given joint configuration.

In [BY94], the authors discuss a biomechanical model of the entire hand encompassing all principal muscles and degrees of freedom. Muscles are modeled by weightless expandable threads. Weightless non-expandable loops surrounding the joints determine the "line-of-action" of muscles. The authors found only the muscles at the thumb and wrist to possess some redundancy, i.e. the same pose can be obtained by several muscle combinations. To overcome this redundancy, muscle effort is minimized.

For evaluation of the prehensile capabilities of the human hand, Buchholz and Armstrong [BA92] proposed a kinematic model based on collision detection between ellipsoids representing the skin surface of the hand segments. Joint flexion angles and skin deformation for power grasp of ellipsoidal objects are predicted and rendered as vector graphics.

The anatomical computer-generated hand model described in [STSL02] consists of bones, tendons, and soft tissue. The latter is modeled by an ellipsoid-shaped mass-spring network at every phalanx, and as an appropriately shaped mass-spring system at the palm. The outer surface of these networks constitutes the skin. Tissue deformation during finger movement is determined using a predictor-corrector method, which also takes into account incompressibility and collision constraints. Tendons are present via their mechanical effects, not geometrically. Their feedback action is modeled through springs opposing joint motion. The fingers are positioned automatically by energy minimization. Although this modeling approach is similar to ours, there are several distinctions: the muscle force model we present is more comprehensive, we model muscles additionally as geometric objects with impact on the shape of the skin, and the triangle mesh we use as skin has been obtained from a range scan of a human hand.

Brand et al. [BBT81] performed measurements of hand and forearm muscles to obtain potential excursion and relative tension of the muscles. Potential excursion is the difference between maximal stretch and maximal contraction of a muscle, i.e. the distance through which a muscle is able to contract actively. They found the potential excursion to be equal to the resting length of the fibers of the muscle. Relative tension denotes the proportional tension of a muscle with respect to the overall amount of possible tension of all studied muscles. These numbers differ far less among individuals and within each individual over time than the absolute strength of a muscle.

Anthropometrical measurements have been carried out by Wagner [Wag88], who extensively measured size and joint mobility of the hands of pianists. He compared his results to studies about other musicians and non-musicians and found that in general piano players have greater mobility in their hands than the average.

2.3.2 Computer Graphics

In computer graphics, hand models have been developed for several typical applications. The most prominent application areas are model-based tracking (see for instance [WH01] for an overview), interactive grasping, and simulation systems, e.g. for surgery planning.
In [OH98], a simple volume-based animatable hand model constructed from geometric primitives is employed for tracking. The model includes anthropometrical and biomechanical constraints: the size of the palm is correlated to the length of the fingers and phalanges. Biomechanical laws determine the valid range and interdependencies of joint motion, thereby reducing the number of degrees of freedom of the model. Heap and Hogg [HH86] have built a statistical hand shape model from simplex meshes fitted to MRI data for their tracking system. For model-based finger motion capturing, Lin et al. [LWH00] employ a learning approach for the hand configuration space to generate natural movement.
A parametric hand model for semi-automatic grasping is described in [MTLD88]. In this model, skinning is based on joint-dependent local deformations, taking into account rounding at joints and bulging. Another approach to grasping is proposed in [GMTT89]. The system uses finite element simulation of the skin and the grasped object in order to simulate both skin and object deformations due to contact. In [RG91], a simple hand model is described that likewise incorporates constraints on the movement range of joints. It was developed for the animation of semi-automatic knowledge-based grasping, where objects are approximated by primitives with individual grasping approach parameters. Another heuristic grasping system was introduced in [ST94]. Objects are stored together with primitives associated with the graspable parts of the object. Grasps are classified depending on the type and size of the primitive, and on the mass of the object. The final position of the hand is determined by inverse kinematics and collision detection. Huang et al. [HBMTT95] extended the previous model. A multi-sensor approach for collision detection has been added, where the sensors are constituted by spheres attached to the joint. Collision detection between hand and object is performed with these sensors to naturally place the hand around the object. Recently, Pollard and Zordan [PZ05] proposed to combine physically based animation with a controller derived from tracking data to obtain animations that capture both active and passive components of grasping. Theoretically, all joints in their model have three degrees of freedom. Joint limits and neutral pose are derived from the motion capture data.
In [KCMT00], artificial intelligence is used to position hand and wrist of a virtual violinist. Finger positions are determined by best-first search, while wrist position and orientation are decided by a neural network. Mulero et al. [MFBLC01] present an anthropomorphic finger model with a tendon transmission system based on pulleys and a position controller. The controller is modeled by a neural network and transforms tendon pull into joint motion. The system can work in an agonist-antagonist fashion. A model of the hand and arms based on manifold mappings was proposed by Kunii et al. [KTM+93]. They also consider inter-joint dependencies. Moccozet et al. [MMT97] use Dirichlet free-form deformations (DFFDs) to simulate the tissue and muscle layer between skin and bones. Muscles are not considered directly, but the use of DFFDs allows the authors to model wrinkles at joints and bulging of segments depending on the angle of rotation of the respective proximal joint. Ip et al. [ICL00] built an anatomy-based hand model with muscles in compliance to [BY94]. The hand is modeled as a collection of hand segments connected by joints, where muscles are weightless expandable threads. Soft tissue, tendons, and ligaments are not modeled explicitly. Given the initial and final hand posture, the system is able to generate in-between states. For describing hand postures, the authors use the Hand Action Coding System [ICL97], a collection of

muscle-based Hand Action Units that encode hand positions. A mathematical model for the complex carpo-metacarpal joint of the thumb based on equations that describe the relationship between tendon excursion and joint angles is described in [Tho81]. The model is matched to experimental data using optimization. Thompson et al. [TBM+88] presented a hand model capable of calculating relative muscle length, distance between pulley point / point of origin and transformed insertion point, moment arm, and moment potential for hand muscles during motion. A wireframe skeleton model obtained from CT scans is rendered together with the tendons, while the single parameters are displayed by bar graphs. Since the system was designed to aid medical doctors in planning tendon transfers, replacement of one musculotendon unit by another can be simulated. In [MTA+01], the joint movements of a hand model composed of rigid bodies are constrained by biomechanical laws. The model was designed for animating American Sign Language. An approach to skinning a hand skeleton using eigendisplacements was proposed in [KJP02]. The resulting hand model can be animated in real-time using graphics hardware. Kurihara and Miyata [KM04] constructed an animatable hand model from CT scans. Animations are obtained through pose space deformation from scans of example poses. Consequently, skin deformations are captured and reproduced realistically. A possible application for the system proposed in [ES03] is guitar playing. From the goal positions for a subset of the finger tips, a hand configuration is computed using inverse kinematics (IK). The poses of the fingers with unspecified tip positions are determined from sample data by interpolating between the k nearest neighbors of the IK solution. Thus, resulting animations exhibit natural joint interdependence. Recently, Tsang et al. [TSF05] developed an accurate biomechanical hand model similar to ours that considers joint interdependencies. The authors solved the control problem by creating animations from keyframes using inverse dynamics.

In addition to the literature on human hand models, several approaches for anatomical modeling and physics-based animation of human faces (cf. Section 2.1) and bodies need to be considered here. In particular, the mass spring system approaches in [TW90, LTW93, LTW95] and the muscle models proposed in [SPCM97, WV97, KHS01] were of interest for our work. Section 3.1.2 deals with the mass spring system and the muscle model by Kähler et al. [KHS01]. Scheepers et al. [SPCM97] model muscles using ellipsoids and bicubic patches, while the muscle model by Wilhelms and Van Gelder [WV97] is based on generalized cylinders. Both muscle models deform in reaction to joint movement.

Concerning the use of feature points for model deformation, the work presented in [KHYS02] should be mentioned: anatomical human head models equipped with feature points can be morphed to obtain head models of new individuals. Based on a 3D scan of the new head, skin mesh and muscles are deformed using radial basis warping functions. From the morphed skin mesh, the deformation function for the skull can be obtained. [SWVG02] proposes a similar approach for articulated creatures. In contrast to [KHYS02], where deformation starts with the skin, this method proceeds from the inside out.

2.4 Tracking Hand and Ball for Baseball Pitches

The pursuit of athletic perfection in baseball has lead to the publication of many textbooks on specific technical aspects [Ste02, Hou00]. In recent years, the athletes' demand for tools to accurately measure and analyze their technical performance has been backed by similar interests from the media. Today, many sports enthusiasts expect concise analysis and visualization of a sports

event during or after the broadcast on TV. In consequence, many researchers have approached baseball from the scientific and technological point of view. The physics of pitching and batting has been thoroughly analyzed in [Ada02]. Alaways examined in his Ph.D. thesis [Ala98] the aerodynamics of a curve-ball. He used a system with ten high-speed video cameras operating at 240 Hz to capture the ball trajectory. Initial flight parameters of the ball were not measured but deduced from the trajectory and a physical model of ball flight. During the Summer Olympics 1996 in Atlanta, Alaways used two 120 Hz high-speed video cameras to track ball positions along the flight trajectory [AMH01]. The K-Zone system [Gue02, Gue03] is technically similar and designed to track the trajectory of a baseball from multiple video streams in real-time using color information and a Kalman filter.

In other popular sports similar systems have been investigated. The LucentVision system [PYOC00] enables tracking of the player positions and the ball trajectory in tennis matches from video images. The ball position is tracked using an algorithm based on ball color and frame-differencing [PJC98]. Rotation axis and spin are not measured. In [DACN02], a modified Circular Hough Transform is used to follow the ball in video broadcasts of a soccer game. Creating images of high-speed motion for analysis of the underlying action has been drawing the attention of researchers for many decades. In the 1870's and 1880's, Eadweard Muybridge conducted his famous experiments to create serial images of fast motion [Muy87]. A setup of twelve cameras was used to capture different stages of a galloping horse. One of the photographs indeed showed the horse with all of its hooves off the ground, corroborating the hypothesis that had led to these experiments. In the 1930's, Harold Edgerton at MIT perfected the use of stroboscope photography to create multi-exposure images of high-speed motion, see for instance [CB94]. However, the acquisition process is usually constrained to actions taking place in a very limited spatial domain for which decent illumination conditions can be set up easily.

In Section 6.3 we will demonstrate that stroboscope photography is not only an appropriate method to accurately track the ball trajectory but also to track the complex articulated motion of the human hand. Many different approaches to tracking articulated human body motion have been investigated in the past, spanning from mechanical over magnetic to optical methods that either rely on optical markers on the human body or set aside any form of intrusion into the scene [AC99, GFG+01]. Commercial optical motion capture systems typically rely on expensive high-framerate video cameras and markers on the body.

Optical approaches for tracking hand articulation usually derive the hand motion from video sequences with the support of an explicit hand model. In [HH86], a point distribution model is used to track hand motion. Stenger et al. [SMC01] employ a kinematic model based on quadric segments and a Kalman filter to determine hand configurations from video. In [WLH01], a 2D cardboard hand representation is used for pose computation. Other approaches that rely on an explicit hand model and image features are the Digiteyes system [RK94] and the work in [Dor93], where colored markers on the hand show the finger joint locations in the video images. A more appearance-based approach is presented in [AS03], where single hand poses are identified via comparison to a database of rendered hand models.

3

Face Models and Animation

The present chapter discusses two face models that serve as foundations for a large part of the work presented in this thesis.
The first section is devoted to MEDUSA, a physics-based human head model capable of real time animation. All animations resulting from the research on non-verbal facial animation as described in Chapter 4 were executed on this platform. MEDUSA heads are modeled after the human anatomy, thus yielding animations of great naturalness. Their anatomical components include skull and jaw, muscles, skin, and connective tissue. The elaborate muscle model and the propagation of muscle deformation to the skin both make the system well suited for animations of subtle facial expressions. Eyes with eyelids, teeth and tongue are represented more simply as geometric objects.
Section 3.2 deals with a learning based approach to facial modeling. From a large database of textured scans of faces, a morphable model is generated which allows faces to be represented as linear combinations of the example scans. The strong point of this technique is the great naturalness and plausibility of all generated faces. Concerning the editing of face models, attributes such as eye shape, skin color, or width of the mouth are learned from the database and then used to modify a face. In addition, it is possible to exchange facial features. Faces on photographs can be reconstructed by the model, thus yielding a three-dimensional face from the two-dimensional image. Conversely, faces from the model can be rendered into images of people, replacing the original faces. All these features make the model the optimal choice for the construction of a facial composite system such as the one from Chapter 5.

3.1 Physics-based Anatomical Models

This section starts with a very brief introduction to the relevant anatomy of the human face. In-depth information can be found in anatomy books as, for example, [PP01, SSZ99]. The following part (Section 3.1.2) describes the individual components of the head models of the MEDUSA animation system by Kähler et al. [KHS01, KHYS02, Käh03]. The implementation of a lip sync algorithm on this platform is explained in Section 3.1.3, and Section 3.1.4 deals with synchronized rendering.

3.1.1 Anatomy of the Face

In terms of bone structure, the human head consists of the moveable mandible and the fixed skull, formed by various closely connected bones. The jaw has four degrees of freedom in speech and mastication [OVBG97]: pitch and yaw as well as horizontal and vertical position,

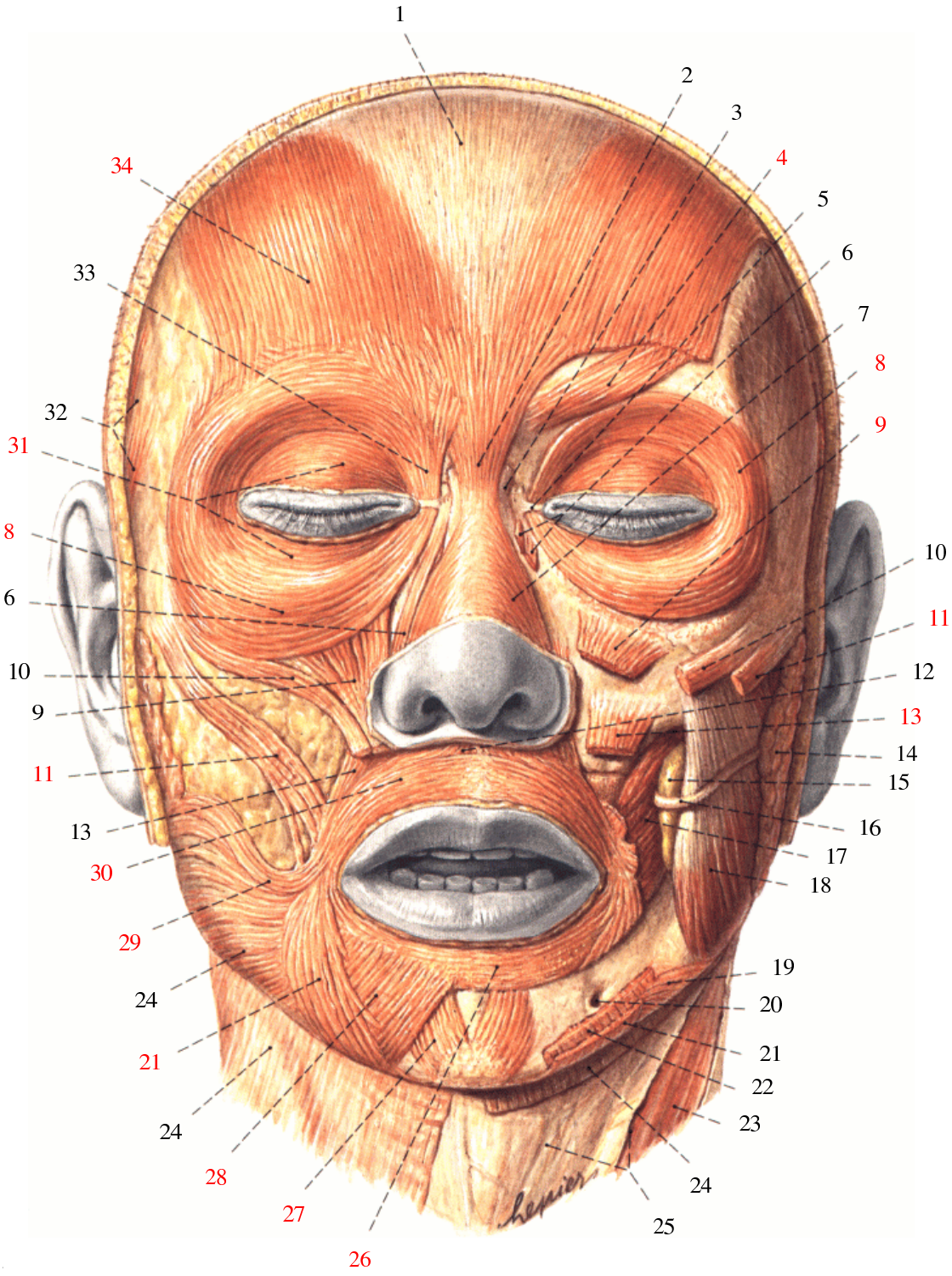


Figure 3.1: **Facial Musculature.** Red numbers correspond to muscles present in MEDUSA models. Numbers refer to Table 3.1. Source: [PP01].

1	epicranial aponeurosis	18	masseter, superficial part
2	nasal bone	19	platysma
3	procerus	20	mental foramen
4	corrugator supercilii	21	depressor anguli oris (triangularis)
5	medial palpebral ligament	22	depressor labii superioris
6	levator labii superioris alaeque nasi	23	sternocleidomastoid
7	nasalis	24	platysma
8	orbicularis oculi, pars orbitalis	25	cervical fascia, investing layer (superficial layer)
9	levator labii superioris	26	orbicularis oris, labial part
10	zygomaticus minor	27	mentalis
11	zygomaticus major	28	depressor labii inferioris (quadratus labii)
12	depressor septi nasi	29	risorius
13	levator anguli oris	30	orbicularis oris, marginal part
14	parotid gland	31	orbicularis oculi, pars palpebralis
15	buccal fat pad	32	temporoparietalis
16	parotid duct	33	depressor supercilii
17	buccinator	34	epicranius, occipitofrontalis, frontal belly

Table 3.1: **Facial Musculature.** Numbers refer to Figure 3.1.

i.e. jaw motion involves a combination of rotation and translation. For speech, motion of the mandible is mostly confined to pitch, i.e. opening and closing of the mouth.

The musculature of the face and especially around the mouth is among the most intricate and complex of the entire human body. The facial muscles interact and work together in mastication, to form expressions and the visual components of speech, to close the eyes in order to keep them wet and to protect them. A total of 82 muscles perform different tasks in the human head and neck [SSZ99]. In Figure 3.1, the muscles of the human face are depicted. Most of them are present by default in the faces of the MEDUSA facial animation system (see Section 3.1.2, especially Figure 3.3).

In the following, the major mimic muscles of the human face are briefly addressed, starting with the muscles covering the brain pan and proceeding downwards towards the neck.
The *occipitofrontalis* and the *temporoparietalis* move the skin of the head. In the case of the *occipitofrontalis*, this involves raising the eyebrows. The *auricularis* muscles allow us to waggle our ears.
Other muscles involved in eyebrow movement are the *depressor supercilii*, which lowers the eyebrows, the *corrugator supercilii*, responsible for pulling the brows together, and the *procerus*, which pulls down the skin over the root of the nose. The *orbicularis oculi* also influences the eyebrows, closes the eyelids and compresses the lacrimal sac.
The *nasalis* muscle dilates the nostrils, while the *depressor septi nasi* pulls the tip of the nose downwards.
The most complex facial muscle is the *orbicularis oris*. It cannot only close the lips, but also make them protrude and retract. Furthermore, it is not a single muscle but is partially composed

of fibers from other muscles. Its deep layer is formed by the *buccinator*, responsible for creating pressure in the mouth for e.g. blowing. The *depressor anguli oris* muscle runs around the lower lip and pulls the angles of the mouth down, while the *levator anguli oris* encircles the upper lip and raises them. The other orofacial muscles like the *zygomatici*, the *levator labii superioris* and the *depressor labii inferioris* merge into this composite structure. The *zygomaticus major* and *zygomaticus minor* raise the upper lip and the angle of the mouth. They create the bump below the eyes that builds during smiling. The *levator labii superioris* and the *depressor labii inferioris* raise the upper lip and pull down the lower lip, respectively. The lips are stretched horizontally by the *risorius* muscle. The *levator labii superioris alaeque nasi* raises the alar wings of the nose together with the upper lip. The *mentalis* and the *transversus menti* move the skin of the chin downward.
The *platysma* finally stretches the skin of the neck.

An account of the inner structure of muscles can be found in Section 6.1.3, and Section 6.1.4 looks at the anatomy and biomechanical properties of human skin.

3.1.2 The MEDUSA System

The physics-based MEDUSA system [KHS01, KHYS02, Käh03] for facial modeling and animation features an underlying anatomical structure consisting of skin, muscles and bones. In addition, its head models comprise eyes with eyelids, a rigid tongue, and teeth.

In order to avoid having to go through the laborious process of assembling the model for every new face from scratch, a generic template is set up once. Given their skin geometry, complete new models can be derived from the generic one. Since the entire anatomical structure is inherited from the template, new faces are instantly animatable, and animation scripts can even be re-used for any given head model.

This section starts with a description of the individual components of the model (see Figure 3.2), followed by a brief account of how the reference head is assembled. Finally, the process of deriving new heads from the template is outlined.

Skin

The skin surface is represented as a triangle mesh. Its biomechanical properties are modeled through a mass spring network, see below.

Skull

The skull is two-part: a rotatable mandible is attached to the cranium. Both components are represented as triangle meshes. The geometry is not used during animation, but only for initialization: the distance between skin and bone is measured, and the lower part of the face is attached to the mandible, so that skin vertices and muscles will follow the jaw rotations.

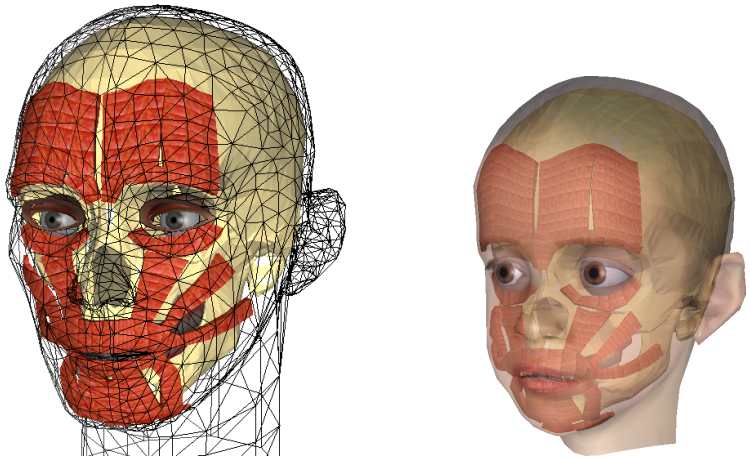
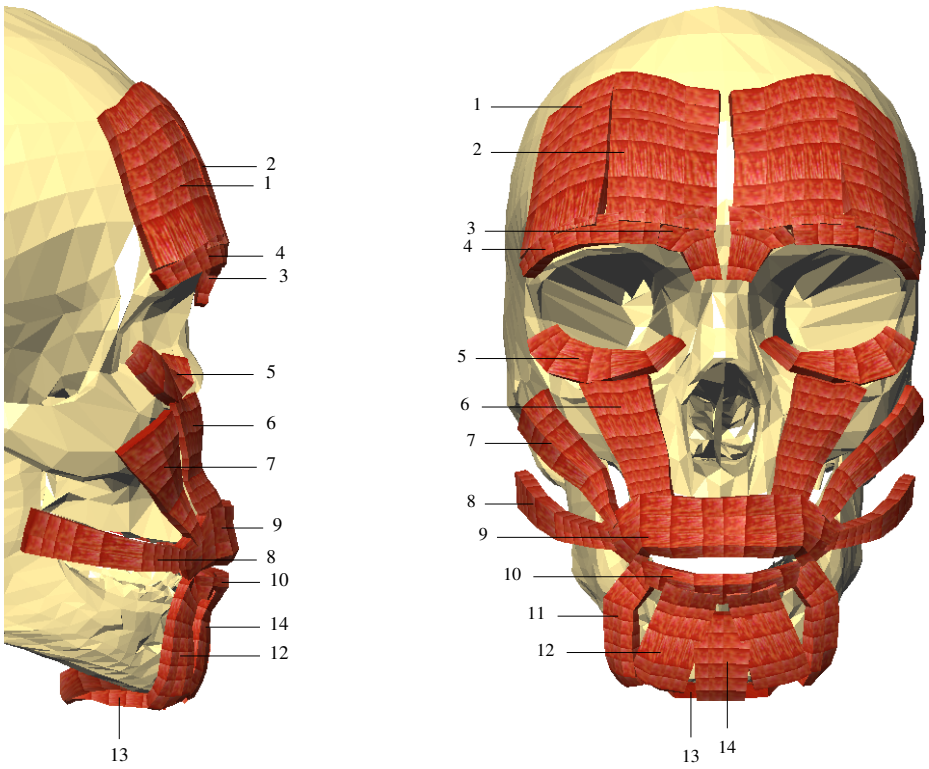


Figure 3.2: **Head structure.** Left: Reference head structure consisting of skull, muscles, skin mesh, eyes, teeth and tongue. Right: Derived head model. The reference head and its components were adapted to the 3D scan of a child to yield a new, animatable head model.

Muscles

Animations are driven by muscle contraction values, specified over time. Muscle contractions occur instantaneously, i.e. internal muscle dynamics or interaction with connective tissue are not taken into account.
The virtual muscles consist of individual, parallel fibers that can either contract linearly for linear muscles, or in a circular fashion for sphincters. The width of a muscle is determined by the number of parallel fibers. Each muscle fiber has a piecewise linear polygon as control structure, with which either ellipsoidal or box like geometric primitives are associated for visualization and attachment to the spring mesh.
Contraction for linear muscles is achieved by moving the control points of the polygon towards the fixed origin of the muscle, while for sphincter muscles, they move towards a specified center of contraction. For the orbicularis oris an axis is declared along which the center of contraction can be moved in order to achieve protrusion and retraction.

Building Muscles. Since the human face is covered by many muscles of complex shape, the process of constructing a set of muscles for a head model is time consuming and requires some amount of flair. Therefore muscles need to be laid out only once for the reference head. When new head models are derived from the template, the muscles are automatically adapted to the new model, i.e. it is instantly animatable (see p. 24).
Muscles are set up interactively by painting a coarse grid onto the skin surface of the generic head model. From this lattice, the muscles are generated automatically and fit between skull and skin by an optimization procedure. The initial grid is made regular by adding control points. Then it is projected onto the face mesh and placed slightly underneath the surface, taking into account skin and average muscle thickness. This is followed by an iterative refinement step, so that the grid does not cut through the skin and that it fulfills the distance constraints to the skin surface. Muscle fibers are generated along the columns of the muscle grid, consisting of one appropriately sized muscle segment per grid cell. The control points are assigned row by row to either skull or jaw depending on the bone the majority of the closest skin vertices are associated with. A muscle is integrated into the spring mesh by attaching the skin nodes in the muscle's



1	occipitofrontalis (outer part)	8	risorius
2	occipitofrontalis (inner part)	9	orbicularis oris (upper part)
3	corrugator supercilii	10	orbicularis oris (lower part)
4	orbicularis oculi, pars orbitalis	11	depressor anguli oris
5	orbicularis oculi, pars palpebralis	12	depressor labii inferioris
6	levator labii superioris	13	mylohyoideus
7	zygomaticus major	14	mentalis

Figure 3.3: **Muscles of the reference head.** This set of muscles is used for lip sync (see Section 3.1.3) and for non-verbal facial animation as described in Chapter 4. Source: [Käh03].

zone of influence to the closest points on the muscle. These attachment points are then mirrored for volume preservation (see p. 22).
A set of muscles is designed only once for the reference head. The muscle grids are transformed along when new faces are derived from the template. From the transformed grids, the muscles for the new head are then set up automatically. The 24 major muscles of facial expression are the default for MEDUSA head models (see Figure 3.3). Additional muscles can easily be added to the system as necessary.

Properties. With the exception of sphincters, which merge into skin on both ends, facial muscles are connected to bone at one end. Their other end either inserts into the skin or is connected to another muscle. Many muscles starting from the mouth region, for example the *zygomatici*, merge into the *orbicularis oris*. Through their interconnectivity, the involved muscles influence each other. In the model, this is achieved by linking the affected control points of the connected muscles by a mass spring network.

This interconnectivity demands that the muscles be able to stretch in addition to their contraction property. To this end, the control points that attach to the bone as well as those that are pulled at by other muscles are kept fixed for each muscle, while intermediate segments are elongated. Unlike muscle contraction, this stretching straightens the muscle out, i.e. it does not follow the original path anymore. Stretching and muscle contraction counteract each other.
When muscles contract, they become thicker, and when they are stretched, they get thinner. For linear muscles, the center exhibits the highest amount of bulging, while sphincter muscles become thicker evenly. Bulging and thinning is propagated to the skin through the mass spring network explained below. Together with the segments, the corresponding spring mesh attachment points, mirrored nodes and nodes on the skin surface are updated. Then the spring mesh simulation is run for one timestep. For visualization purposes, the segment shape is adjusted as well.
In contrast to the general behavior, where all fibers of a muscle contract in the same way, fibers belonging to the orbicularis oris are controlled independently. This allows for lip protrusion and retraction.

Muscle Deformation. Muscle contraction values are provided directly by the user. They lead to active shortening and bulging, or to passive stretch by other muscles and thinning. These deformations are propagated via the mass spring network towards the skin.
For active contraction, the muscle control points are moved according to the contraction value. In the case of a linear muscle, the control points are constrained to follow the original course of the muscle. Sphincter muscles contract towards their center. The *orbicularis oris* has an additional parameter for protrusion and retraction along a user specified axis. Hereby, the individual muscle fibers protrude / retract with differing intensity: the outer part of the orbicularis oris does so to a lesser extent than the inner part. This is achieved by gradually shifting the center of contraction for the muscle fibers along the axis. For the outer fibers, it remains in the plane of the muscle, while for the innermost part, it is moved to the point on the axis specified by the parameter value.
The only other method of moving a muscle is by jaw rotation. The jaw is animated through a direct parameter. When it rotates, muscles attached to the jaw are moved along by rotating their control points.
After deforming the muscle fibers according to their contraction value or to jaw rotation, the muscle connection constraints must be satisfied. Muscle connections are preserved by restoring the original distance of the involved control points to their center. The center is defined as the weighted average of the control points, where points belonging to a muscle with higher contraction value have an increased weight, so that the center is pulled towards these control points. Thereby the system grants greater influence to more strongly contracted muscles. In order to propagate the resulting local deformations, these geometric modifications are followed by a simulation of the muscle spring mesh.
Then the muscle shapes are adjusted to exhibit bulging and stretching. Finally, the spring mesh is updated.

Mass Spring Network

Skin, muscles and skull are connected by a mass spring network. This approach was chosen because it approximates the biomechanical properties of skin and connective tissue well, while at the same time permitting real time animation.
The spring mesh consists of nodes $\mathcal{N} = \{n_i \,|\, 0 \leq i < N\}$ with point masses $\mathcal{M} = \{m_i \,|\, 0 \leq$

$i < N\}$, that are connected by Hookean springs $\mathcal{S} = \{s_j \,|\, 0 \leq j < M\}$. When mass nodes are displaced, the connecting springs are stretched. The so-caused strain forces other nodes to move, until finally a new equilibrium is reached, where the forces acting on each node cancel out. The forces affecting a node can be computed as explained in the following.
Consider a spring $s_j \in \mathcal{S}$ which connects two nodes $n_s, n_e \in \mathcal{N}$ at positions $\mathbf{x}_s, \mathbf{x}_e \in \mathbb{R}^3$. The spring's current length is $l_j = \|\mathbf{x}_e - \mathbf{x}_s\|$. Its rest length l_j^0 denotes the spring's length in a relaxed state of the system, and its stiffness constant $c_j \in \mathbb{R}$ describes its elasticity. The force $\mathbf{F}_{j,e}$, which s_j exerts on its end node n_e, is then:

$$\mathbf{F}_{j,e} = c_j \frac{l_j - l_j^0}{l_j}(\mathbf{x}_e - \mathbf{x}_s) \ .$$

The total force $\mathbf{F}_i$ at node n_i is the sum of forces exerted by springs ending in n_i minus the sum of forces exerted by springs originating from n_i:

$$\mathbf{F}_i = \sum_{\substack{s_j \in \mathcal{S} \\ s_j \text{ ends in } n_i}} \mathbf{F}_{j,i} \quad - \sum_{\substack{s_j \in \mathcal{S} \\ s_j \text{ starts in } n_i}} \mathbf{F}_{j,i} \ .$$

Now let $\mathbf{x}_i$ be the position of node n_i. Velocity and acceleration of n_i are $\dot{\mathbf{x}}_i$ and $\ddot{\mathbf{x}}_i$. The system of equations of motion for n_i can be formulated as

$$m_i \ddot{\mathbf{x}}_i + \gamma \dot{\mathbf{x}}_i - \mathbf{F}_i - \mathbf{F}_{\text{ext}} = 0 \ ,$$

with $\mathbf{F}_{\text{ext}}$ denoting the sum of all external forces acting on n_i. γ is a damping factor, and the damping term $\gamma \dot{\mathbf{x}}_i$ models energy loss due to friction.
The equations of motion are second-order ordinary differential equations. Initial positions and velocities are known for all nodes. To obtain an animation and hence solve this initial value problem for the animation time steps, Euler integration is used for the first time step, and from then on the leapfrog Verlet method [AT89] is employed.

The spring mesh is set up automatically from the geometry. Nodes and edges of the skin mesh are converted into mass points and springs to model epidermis and dermis. The stiffness constant of the skin springs is biphasic to account for the stress-strain behavior of real skin. On bones and muscles, static mass points are inserted and connected to the skin nodes. The stiffness value of the connective springs is rather low to model subcutaneous fatty tissue which allows the skin to slide freely over muscles and skull.
Volume preservation to prevent the skin from penetrating muscles and skull is achieved by attaching a spring to each skin mesh node that pulls the mass point outwards. This additional spring mirrors the spring which connects the node to the underlying muscle or bone (see Figure 3.4). The force exerted by these outward facing springs can be interpreted as modeling the internal pressure of the skin cells. This method reliably prevents penetration, except for very violent movements.

Eyes, Teeth, Tongue

The eyes together with the animatable eyelids, teeth and tongue are modeled as rigid geometric objects. These components are not animated through muscles, but via direct parameters.

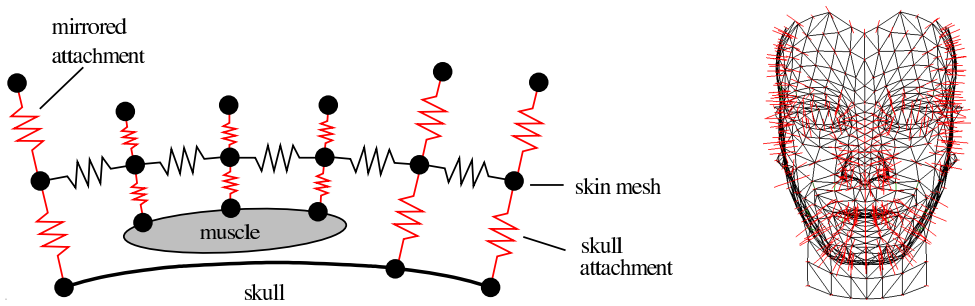


Figure 3.4: **Mass spring network.** Left: Springs connect the skin surface to skull and muscles; attachments are mirrored outwards. Right: spring mesh of the reference head. Black springs: surface edges; red springs: true and mirrored attachments. Source: [Käh03].

Figure 3.5: **Animation with wrinkles.** Two different facial expressions with automatically generated expressive wrinkles, rendered at 100 fps using hardware bump mapping.

Expressive Wrinkles

The degree of realism of facial animations can be increased significantly by including expressive wrinkles with variable intensity. MEDUSA makes use of the *vertex program* and *register combiners* extensions of the NVidia GeForce6 graphics board to render bump mapped wrinkles at real-time frame rates. The bump map for the wrinkles is created from a "wrinkle height field" [Wyn01], which is in turn generated from the layout of the expressive wrinkles in the skin texture. The intensity of the wrinkles is controlled by the contraction values of the corresponding muscles. Contracting, for instance, the *frontalis* muscle, which is responsible for frowning, automatically results in wrinkles appearing on the forehead. Figure 3.5 shows two examples of different facial expressions with automatically generated expressive wrinkles.

Building the Reference Head

After having obtained triangle meshes for the skin and the skull, both meshes are interactively equipped with anthropometric landmarks. The bones are then transformed to fit the skin geometry using these landmarks and a radial basis warping function. Measurement of the skin to skull distance is performed automatically, as is the attachment of the appropriate skin vertices to either the jaw or the skull. Muscles are laid out interactively, and eyes, teeth and tongue are generated procedurally and added to the model. Finally, the connective mass spring network is set up automatically to yield the animatable model.

The advantage of this approach is that it starts from the known outside and then proceeds to construct a matching anatomical structure. After the template has been assembled, new head models can be derived from it easily.

Creating New Animatable Head Models from the Template

A fully animatable model can be derived for any face where the skin geometry is available. This is done by deforming all parts of the template head model to adapt it to the target head. To this end, the reference head and skull are equipped with landmarks at anthropometrically meaningful positions. For the deformation, feature points must be placed at corresponding locations on the target skin mesh. From the correspondence between the skin landmarks on the source and target geometry, a radial basis warping function is set up to transform the template skin mesh to match the new head geometry.
The skull landmarks are related to their counterparts on the skin by an offset. From this offset and the new positions of the skin feature points, the target skull landmark positions can be obtained, allowing to set up the warping function for the skull.
To transform the muscles, the same deformation function as for the skin is applied to the muscle grids. After that, the muscles for the target head are computed from the grid.
The remaining parts of the face, i.e. the geometric objects for eyes, teeth and tongue, can only be positioned and scaled automatically, the fine tuning needs to be done by hand.

3.1.3 MEDUSA Rises to Speak

Lip sync is the ultimate test for every facial animation system due to the high complexity of mouth movements and the required exactness, both in terms of timing and in terms of configuration. Although the majority of people is not proficient in lipreading, they are mercilessly observant and detect even slight misalignments between audible and visible speech. Actual lip position seems to be less important.

In order to be convincing, the animation of lip movement for speech must consider *coarticulation*, i.e. the coloring of speech segments by neighboring phonemes. This phenomenon does not only play a role at the audible level, but greatly affects lip shape during speech.
Within MEDUSA, the coarticulation model by Cohen and Massaro [CM93] was implemented for speech synchronization. As input it requires the phonemes of the desired utterance and their durations. From these, the trajectories for the animation parameters over time are computed.
After an overview of the approach follows an account of how we adapted the method to work with the muscle-based MEDUSA facial animation system. Synchronized real-time rendering of audio and animations is described in Section 3.1.4.

The Coarticulation Model by Cohen and Massaro

Cohen and Massaro [CM93] adopted the articulatory gesture model by Löfqvist [Löf90]. The special feature of gesture-based models [Koh92] is that they do not use phonemes as basic phonological unit but gestures. This extricates them from the problem to map discrete, context-free segments onto a context-laden continuous signal stream. The structure of an utterance is given by the coordination of gestures instead of by a string of phonemes. Gestures and phonemes can

be regarded as the high- and low-level representation of the same system: each segment is associated with the gestures that are executed during the realization of the segment. This makes a conversion between the two levels of representation comparatively easy.
Coarticulation means that segments influence one another. This happens via gestures that belong to one phoneme and that reach into other segments, i.e. coarticulation is the result of overlapping gestures. Therefore during speech at any point in time the vocal tract is shaped by several gestures that belong to different segments. Overlapping gestures of successive segments show blends and aggregations into a single gesture.
While the gestures of a segment are active, they shape the vocal tract together with the gestures of the neighboring segments; i.e. a segment influences the vocal tract during a certain time interval, possibly in conjunction with other segments. This time-varying influence of the segment over the vocal tract is called *dominance*. The dominance of a segment may differ between articulators.

From this model, Cohen and Massaro derived a method to compute articulator behavior during speech that includes coarticulation. A dominance function for every facial parameter - phoneme pair describes the influence of the segment on the behavior of the animation parameter at any point in time. The function is used as a weight in determining how close a parameter gets to reaching its goal position and which position it takes at a given time. The dominance of a segment does not automatically cease at the segment boundary but can well reach into other segments. Thus, dominance functions of neighboring segments may overlap.
The behavior of a facial control parameter p over time can be determined as follows: if $D_{s,p}$ is the function describing the dominance of segment s over p, and $T_{s,p}$ is the goal position of parameter p for the phoneme s, the function describing the behavior F_p of parameter p over time is given as the weighted average of the targets of p during the whole utterance:

$$F_p(t) = \frac{\sum_{s=1}^{N} D_{s,p}(t) T_{s,p}}{\sum_{s=1}^{N} D_{s,p}(t)} \ ,$$

where N is the number of segments in the utterance.
As dominance functions, Cohen and Massaro propose the negative exponential functions

$$D_{s,p}(t) = \begin{cases} \alpha_{s,p} e^{-\theta_{\leftarrow s,p} |\tau(t)|^c} & \text{if } \tau(t) \geq 0 \\ \alpha_{s,p} e^{-\theta_{\rightarrow s,p} |\tau(t)|^c} & \text{if } \tau(t) < 0 \ . \end{cases}$$

The parameter $\alpha_{s,p}$ determines the magnitude of the dominance function. It describes how susceptible a parameter is to coarticulation, i.e. how close the facial parameter comes to reaching its target position. The rate at which the function rises and falls is given by θ. This parameter can have different values for increase and decrease, allowing for differences in forward and backward coarticulation. $\tau(t)$ describes the time distance from the function peak. For time t, $\tau(t)$ is defined as

$$\tau(t) = (t^s_{\text{start}} + \frac{t^s_{\text{dur}}}{2}) + t^{s,p}_{\text{off}} - t \ .$$

t^s_{start} indicates the starting time of segment s and t^s_{dur} is its duration, i.e. $t^s_{\text{start}} + \frac{t^s_{\text{dur}}}{2}$ is the center of s. As the peak of the dominance function of s over facial control parameter p need not necessarily lie at the segment's center, a parameter $t^{s,p}_{\text{off}}$ describing the time offset from the center to the peak of domination can be specified. Variations in parameter c change the characteristics of the transition between adjacent segments. When c increases, the values for the articulators are more likely to hit their goal positions while at the same time there is an overall decrease in coarticulatory effects. Moreover, transitions between segments become more abrupt. Cohen and Massaro recommend to choose $c = 1$.

Adaptation and Extensions

The following paragraphs hold a brief description of the integration of the coarticulation technique into the MEDUSA system [AHS02b].

Muscles. For visual speech, we found it sufficient to use the following muscles (see Figure 3.3):

- *orbicularis oris* upper lip and lower lip (encircle the mouth)
- *mentalis* (raises the chin towards the lips)
- *risorius* left and right (move the corners of the mouth towards the ears)
- *depressor labii inferioris* left and right (pull down the lower lip).

The orbicularis oris and the jaw play the most important role in the animations of speech, which makes the flexibility of MEDUSA's orbicularis oris model a definite advantage.
Adapting the original coarticulation algorithm to our muscle-based approach is straightforward: the facial control parameters of the parameterized face model employed by Cohen and Massaro can be replaced by muscle contraction, jaw and tongue parameters directly.

Speed-up. The greatest known expansion for coarticulation effects is eleven segments. Yet the original algorithm by Cohen and Massaro considers the effects on all segments in the whole utterance. Especially for longer pronouncements stretching over several sentences this produces unnecessary computational overhead. Therefore in our implementation, the algorithm only considers the eleven following and preceding segments of a phoneme. In the twelfth segment, the function is led to 0 using cubic Hermite interpolation in order to ensure continuity.

Closure and Release Phases. For the production of the bilabial stops /p/, /b/ and the nasal /m/, it is vital that the lips are fully closed. For the fricatives /f/ and /v/, the lower lip must touch the teeth. That these target positions are reached exactly is not only important for the production of the sound but also for its visual perception. For the stops /b/, /p/, the lips are held closed for a certain (usually very short) time interval, the *closure* phase. During the following *release*, the lips burst open to let the retained air rush out. Although the closure can be as short as 5 msec, it is used as a cue during speech. With the two above mentioned fricatives it is similar. However, here the "closure" is not complete, the air is pressed through the space between the teeth. Therefore, a release phase does not exist. For the production of /m/, the lips are fully closed but the velum is lowered, thus opening the connection between the oral and nasal cavity. Through this passage, the air can escape, again obviating the need for a release phase.
Our system models the closure and release phase of the stops /b/, /p/ separately. The fricatives /f/ and /v/, and the nasal /m/ are handled in the same way as the bilabial closure. This is possible because we do not take into account air pressure.
Animation frames are generated at uniform intervals. At the beginning of the closure of a phoneme, however, a key frame is generated, even if this disturbs the sampling rhythm. By assigning a high magnitude to the dominance function, the lips come sufficiently close to their targets. The next key frame is computed at the beginning of the release phase, if existent, or of the next phoneme. Here, the normal procedure with equidistant time steps is resumed.
The peak of the dominance function of the release visemes is always set to the beginning of the corresponding interval.

3.1.4 Synchronized Rendering

In a real-time setting, it is important to achieve not only high rendering frame rates, but also accurate synchronization to audio [AHK+02]. In the MEDUSA system, the animation is generated by a physics-based simulation, running in its own thread on a dual processor PC. This simulation thread performs numerical integration of the equations of motion for the mass-spring network representing the facial skin layer. The displacements of skin mesh vertices for one simulation time step are stored as a *simulation key frame* in a buffer along with the current simulation time, measured in wall-clock time. We typically obtain simulation frame rates of about 40 key frames per second. The second thread on the other CPU is responsible for rendering. Here, successive simulation key frames are interpolated according to the current rendering time, which is also measured in wall-clock time.
Simulation and audio are running at the same speed, because animation parameters and audio are generated from the same phonetic description. Synchronization is thus achieved by initiating the audio output when the first frame is rendered. Due to stable rendering frame rates of about 100 fps, we obtain a high consonance between audible speech and rendered images with a maximum inaccuracy of about 10 milliseconds.

3.2 A Photorealistc Modeling Tool for Faces

For the facial composite system (Chapter 5), we relied heavily on the work by Blanz et al. [BV99, BSVS04]. The strength of their approach lies in the fact that it is learning-based. A database of 200 textured 3D scans of mostly Caucasian faces (100 male, 100 female) is used to construct a morphable model of faces [BV99]. On the one hand, any Caucasian face can be described by the model, and on the other, faces generated from the model are realistic and natural.

3.2.1 Spanning Face Space with a Morphable Model

The example faces in the database consist of textured 3D meshes obtained from laser scans. Faces were normalized with respect to position and orientation and brought into full correspondence.

A face can now be represented by a shape vector $\mathbf{s}$ and a texture vector $\mathbf{t}$ defined as follows:

$$\begin{aligned} \mathbf{s} &= (x_1, y_1, z_1, \ldots, x_n, y_n, z_n) \\ \mathbf{t} &= (r_1, g_1, b_1, \ldots, r_n, g_n, b_n) \,, \end{aligned}$$

where x_i, y_i, z_i denote the position of vertex v_i and r_i, g_i, b_i are the corresponding texture values. The number of vertices is n. This makes $\mathbf{s}$ simply the concatenation of the 3D coordinates of all mesh vertices, and $\mathbf{t}$ the concatenation of the corresponding entries in the texture map.
Doing this for N faces yields a set of shape vectors $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ and a set of texture vectors $T = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$. Because all faces are in correspondence, new face shapes and textures can now be obtained through linear combinations of elements from S and T:

$$\begin{aligned} \mathbf{s}_{\text{new}} &= \sum_{k=1}^{N} a_k \mathbf{s}_k \quad \text{with} \quad (a_i)_{i=1}^{N} \in \mathbb{R}^N \\ \mathbf{t}_{\text{new}} &= \sum_{k=1}^{N} b_k \mathbf{t}_k \quad \text{with} \quad (b_i)_{i=1}^{N} \in \mathbb{R}^N \,. \end{aligned}$$

The set of all textured faces within this vector space constitutes the morphable model. Faces will be realistic as long as they are not too many standard deviations away from the average. Average shape $\bar{\mathbf{s}}$ and average texture $\bar{\mathbf{t}}$ can be obtained by setting $a_k = b_k = \frac{1}{N}$ for $k = 1, \ldots, N$.

Now let

$$\begin{aligned} \Delta\mathbf{s}_k &= \mathbf{s}_k - \bar{\mathbf{s}} \\ \Delta\mathbf{t}_k &= \mathbf{t}_k - \bar{\mathbf{t}} \end{aligned}$$

for $k = 1, \ldots, N$ denote the difference between a face and the average. These characteristic vectors describe what makes each face unique. Performing Principal Component Analysis (PCA) on the characteristic vectors will yield an orthonormal base for the face space. Carrying out PCA is equivalent to diagonalizing the covariance matrices $\mathbf{C_s}$ for shape and $\mathbf{C_t}$ for texture. Let $\Delta\mathbf{S}$ be the matrix with column vectors $\Delta\mathbf{s}_i$ and $\Delta\mathbf{T}$ the corresponding texture data matrix. Then $\mathbf{C_s}$ and $\mathbf{C_t}$ are defined as follows:

$$\begin{aligned} \mathbf{C_s} &= \frac{1}{N}\Delta\mathbf{S}\Delta\mathbf{S}^T \\ \mathbf{C_t} &= \frac{1}{N}\Delta\mathbf{T}\Delta\mathbf{T}^T \,. \end{aligned} \tag{3.1}$$

The desired diagonalizations

$$\begin{aligned} \mathbf{C_s} &= \mathbf{U_s}\text{diag}(\sigma_i^2){\mathbf{U_s}}^T \\ \mathbf{C_t} &= \mathbf{U_t}\text{diag}(\tau_i^2){\mathbf{U_t}}^T \,, \end{aligned} \tag{3.2}$$

can be computed from the singular value decompositions of $\Delta\mathbf{S}$ and $\Delta\mathbf{T}$:

$$\begin{aligned} \Delta\mathbf{S} &= \mathbf{U_s}\mathbf{W_s}{\mathbf{V_s}}^T \\ \Delta\mathbf{T} &= \mathbf{U_t}\mathbf{W_t}{\mathbf{V_t}}^T \,. \end{aligned} \tag{3.3}$$

The columns of $\mathbf{U_s}$ and $\mathbf{U_t}$ are the principal components, i.e. the eigenvectors $\mathbf{s}_i^*$ of $\mathbf{C_s}$ and $\mathbf{t}_i^*$ of $\mathbf{C_t}$, arranged in decreasing order of their corresponding eigenvalues. They constitute the new base. Hereby, $0 < i < N$, because we operate on the linearly dependent $\Delta\mathbf{s}_i$ and $\Delta\mathbf{t}_i$. $\mathbf{W_s} = \sqrt{N}\text{diag}(\sigma_i)$ and $\mathbf{W_t} = \sqrt{N}\text{diag}(\tau_i)$ yield the standard deviations σ_i and τ_i along the principal components. $\mathbf{V_s}$ and $\mathbf{V_t}$ are orthogonal matrices.

Now we can describe every face in terms of the new bases:

$$\begin{aligned} \mathbf{s}_{\text{new}} &= \bar{\mathbf{s}} + \sum_{i=1}^{N-1} \alpha_i \mathbf{s}_i^* \quad \text{with} \quad (\alpha_i)_{i=1}^{N-1} \in \mathbb{R}^{N-1} \\ \mathbf{t}_{\text{new}} &= \bar{\mathbf{t}} + \sum_{i=1}^{N-1} \beta_i \mathbf{t}_\mathbf{i}^* \quad \text{with} \quad (\beta_i)_{i=1}^{N-1} \in \mathbb{R}^{N-1} \,. \end{aligned}$$

The nice thing is that the eigenvalues are an indicator of the contribution of the corresponding base vectors in terms of variance. Using only the first 149 principal components (or base vectors) for texture and shape is sufficient [BSVS04]. The other components describe noise and other variations that are not face specific. This makes PCA well suited for compression.

$-3\Delta\mathbf{s}_A$ $-3\Delta\mathbf{t}_A$ | $-\Delta\mathbf{s}_A$ $-\Delta\mathbf{t}_A$ | original | $+\Delta\mathbf{s}_A$ $+\Delta\mathbf{t}_A$ | $+3\Delta\mathbf{s}_A$ $+3\Delta\mathbf{t}_A$

Figure 3.6: **Facial attributes.** Center: original face. Right: after adding the shape and the texture vectors for attribute A = *heart shape* 1x and 3x, respectively. Left: after subtracting the shape and the texture vectors for attribute A 3x and 1x, respectively. This pushes the face in the direction opposite to *heart shape*, i.e. the face becomes more pear shaped.

The ugly thing is that modifications using principal components are very unintuitive, since they operate on the entire face. While there is a principal component responsible for e.g. mouth width, it is impossible to describe the effects of other components. For a lot of facial characteristics such as obesity there simply is no principal component. Therefore some alternative is required if the system is to be used for modeling faces. The solution proposed by Blanz et al. [BV99, BAHS] were *facial attributes*.

3.2.2 Facial Attributes as an Intuitive Means to Modify Faces

Arbitrary facial attributes can be described by shape and texture vectors in face space. By adding multiples of these vectors to (subtracting them from) a face, the face will be altered to show the attribute (its opposite) to the desired degree. All other properties of the face will remain unchanged.
For an attribute A, the vectors $\Delta\mathbf{s}_A$ and $\Delta\mathbf{t}_A$ are obtained as follows: first weights w_k are assigned manually to all example faces F_k, $k = 1, \ldots, N$, according to how salient A is in F_k. The vectors themselves are then computed as:

$$\begin{aligned} \Delta\mathbf{s}_A &= \frac{1}{N}\sum_{k=1}^{N} w_k \Delta\mathbf{s}_k \\ \Delta\mathbf{t}_A &= \frac{1}{N}\sum_{k=1}^{N} w_k \Delta\mathbf{t}_k \ . \end{aligned} \tag{3.4}$$

The result of applying both the shape and texture vector of attribute A = *heart shape* with varying intensities to a face is shown in Figure 3.6. Remarkably, the individual features of the face are only marginally affected, when the overall face shape is modified. See below for the theoretical background.

A face F with shape $\mathbf{s} = \bar{\mathbf{s}} + \sum_{k=1}^{N-1} \alpha_k \mathbf{s}_k^*$ and texture $\mathbf{t} = \bar{\mathbf{t}} + \sum_{k=1}^{N-1} \beta_k \mathbf{t}_k^*$ can be caricatured by increasing its distance from the average face. For a given caricature level c, the new shape

and texture vectors are

$$\begin{aligned} \mathbf{s}_c &= \bar{\mathbf{s}} + c \cdot \sum_{k=1}^{N-1} \alpha_k \mathbf{s}_k^* \\ \mathbf{t}_c &= \bar{\mathbf{t}} + c \cdot \sum_{k=1}^{N-1} \beta_k \mathbf{t}_k^* \, . \end{aligned}$$

Note that caricatures, as opposed to attributes, are computed in principal component representation.

In order to increase the expressiveness of the model, the face can be divided into subregions. This is equivalent to operating on a subspace where vector entries are 0 for those vertices that do not belong to the specific segment. Operations are then executed independently on subregions that are afterwards blended at their boundaries. This is useful when one wants, for example, to modify the mouth, but does not want the remainder of the face to change.

Mathematical Background

This paragraph delves into the mathematical background of navigating face space by attribute manipulation. Without loss of generality, we assume that the means of the $\Delta\mathbf{s}_i$, the $\Delta\mathbf{t}_i$, and the w_i are zero. For the sake of simplicity, only the case of the $\Delta\mathbf{s}_i$ is considered, but all observations apply analogously to the $\Delta\mathbf{t}_i$.
We need to estimate the function f that assigns attribute values to faces. Following the gradient ∇f of this function achieves the desired change in attribute with a minimal change in facial appearance. Given the limited set of data, we choose a linear regression[1] for f.
The coefficients of the linear regression depend on the scalar product. As a consequence, the choice of the scalar product also affects the gradient of f, which specifies the rate and direction of greatest change of the function. As demonstrated at the end of this paragraph, an appropriate choice in our setting is the scalar product $\langle\cdot\rangle_M$ derived from the Mahalanobis distance. In this distance measure, which is adapted to the probability density estimated by PCA, distances are measured relative to the standard deviation observed in the dataset of examples. The resulting functional f is

$$\begin{aligned} f(\Delta\mathbf{s}) &= \langle \Delta\mathbf{s}, \Delta\mathbf{s}_A \rangle_M \\ &= \langle \Delta\mathbf{s}, {\mathbf{C}_\mathbf{s}}^{-1} \Delta\mathbf{s}_A \rangle \end{aligned}$$

for an attribute A.
The regularization problem is then a least-squares minimization

$$\begin{aligned} E &= \sum_{i=1}^{N} (\langle \Delta\mathbf{s}_i, \Delta\mathbf{s}_A \rangle_M - w_i)^2 \\ &= \| \Delta\mathbf{S}^{\mathrm{T}} {\mathbf{C}_\mathbf{s}}^{-1} \Delta\mathbf{s}_A - \mathbf{w} \|^2 \\ &= \min \end{aligned}$$

[1] *Linear regression* approximates the relationship between an output variable $y \in \mathbb{R}$ and a set of input variables $x_1, \ldots, x_n \in \mathbb{R}$, $n \in \mathbb{N}$, by fitting a straight line through the data, i.e. $y = a_0 + a_1 x_1 + \ldots + a_n x_n + e$, where the error e is minimal in the least squares sense and has mean 0. Using this mapping, it is possible to predict the value of y for new values of the input variables $x_1, \ldots, x_n$.

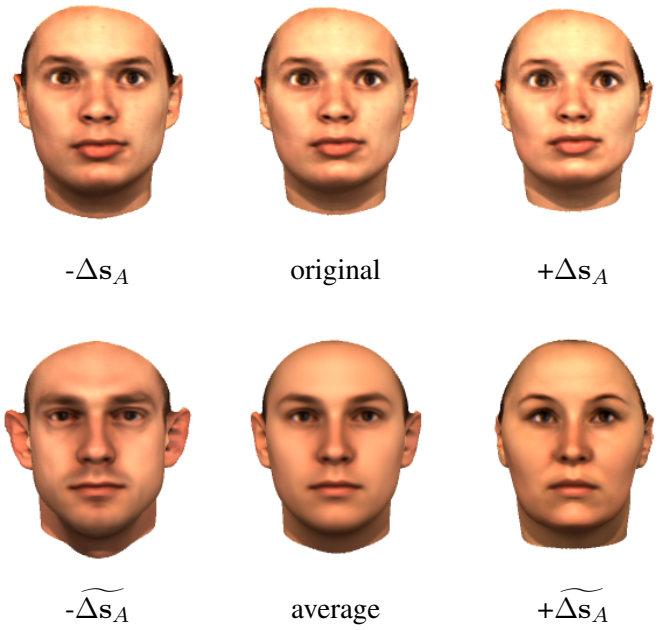


Figure 3.7: **Comparison of Mahalanobis distance and L_2-norm.** Top row: by adding multiples of the attribute vector $\Delta\mathbf{s}_A$ to a face, attributes such as gender can be changed. Characteristics such as the shape of the mouth are retained. Bottom row: simply adding the gradient $\nabla f = \widetilde{\Delta\mathbf{s}_A}$, which defines the steepest slope of the attribute in L_2-norm, would alter the individual features.

with the data matrix $\Delta\mathbf{S}$ and $\mathbf{w} = (w_1, ..., w_N)$. Using the singular value decomposition from Equation (3.3), as well as Equation (3.2), we obtain

$$\begin{aligned} E &= \left\| \left(\mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}}\right)^{\mathrm{T}} \left(\frac{1}{N}\mathbf{U}\mathbf{W}^2\mathbf{U}^{\mathrm{T}}\right)^{-1} \Delta\mathbf{s}_A - \mathbf{w} \right\|^2 \\ &= \left\| N \cdot \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^{\mathrm{T}}\Delta\mathbf{s}_A - \mathbf{w} \right\|^2 . \end{aligned}$$

The optimal solution can be found using the pseudo-inverse, which is easy to compute here since the problem is already decomposed into orthogonal and diagonal matrices[2]:

$$\begin{aligned} \left(N \cdot \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^{\mathrm{T}}\right)^{+} &= \frac{1}{N}\mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \\ &= \frac{1}{N}\Delta\mathbf{S} , \end{aligned}$$

which leads to the expected result from Equation (3.4):

$$\begin{aligned} \nabla f = \Delta\mathbf{s}_A &= \frac{1}{N}\Delta\mathbf{S}\mathbf{w} \\ &= \frac{1}{N}\sum_{i=1}^{N} w_i \Delta\mathbf{s}_i , \end{aligned}$$

[2]For a matrix $\mathbf{M} \in \mathbb{R}^n$ with $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{R}^{\mathrm{T}}$ ($\mathbf{L}$, $\mathbf{R}$ orthogonal, $\mathbf{D}$ diagonal), the pseudo-inverse $\mathbf{A}^+$ can be computed as $\mathbf{A}^+ = \mathbf{R}\mathbf{D}^{-1}\mathbf{L}^{\mathrm{T}}$.

i.e. the weighted sum of the input vectors. The direction $\Delta\mathbf{s}_A$ defines the steepest ascent or descent in terms of Mahalanobis distance. Adding a vector $\frac{N}{\|\mathbf{w}\|^2}\Delta\mathbf{s}_A$ to $\Delta\mathbf{s}$ changes $f(\Delta\mathbf{s})$ by a value of 1:

$$\begin{aligned} f(\Delta\mathbf{s} + \frac{N}{\|\mathbf{w}\|^2}\Delta\mathbf{s}_A) &= f(\Delta\mathbf{s}) + \frac{N}{\|\mathbf{w}\|^2}\langle\Delta\mathbf{s}_A, {\mathbf{C_s}}^{-1}\Delta\mathbf{s}_A\rangle \\ &\overset{(3.4,3.1)}{=} f(\Delta\mathbf{s}) + \frac{N}{\|\mathbf{w}\|^2}\langle\frac{1}{N}\Delta\mathbf{S}\mathbf{w}, (\frac{1}{N}\Delta\mathbf{S}\Delta\mathbf{S}^T)^{-1}\frac{1}{N}\Delta\mathbf{S}\mathbf{w}\rangle \\ &\overset{(3.3)}{=} f(\Delta\mathbf{s}) + \frac{1}{\|\mathbf{w}\|^2}\langle\mathbf{U}\mathbf{W}\mathbf{V}^T\mathbf{w}, \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T\mathbf{w}\rangle \\ &= f(\Delta\mathbf{s}) + 1\ . \end{aligned}$$

The result of adding multiples of $\Delta\mathbf{s}_A$ to faces is shown in Figure 3.6 and in the top row of Figure 3.7.
In order to motivate the use of the scalar product $\langle\cdot\rangle_M$, consider the same problem with the standard scalar product, which will lead to a vector $\widetilde{\Delta\mathbf{s}_A}$ different from the previous solution:

$$\begin{aligned} f(\Delta\mathbf{s}_i) &= \langle\Delta\mathbf{s}_i, \widetilde{\Delta\mathbf{s}_A}\rangle \\ E &= \|\Delta\mathbf{S}^{\mathrm{T}}\widetilde{\Delta\mathbf{s}_A} - \mathbf{w}\|^2 \\ &= \min\ . \end{aligned}$$

This can be solved using the pseudo-inverse $\Delta\mathbf{S}^{\mathrm{T}+}$ of $\Delta\mathbf{S}^{\mathrm{T}}$:

$$\widetilde{\Delta\mathbf{s}_A} = \Delta\mathbf{S}^{\mathrm{T}+}\mathbf{w}\ .$$

For manipulating attributes of faces, we would then add multiples of the gradient $\nabla f = \widetilde{\Delta\mathbf{s}_A}$. This vector achieves the desired change in $f(\Delta\mathbf{s})$ with minimal effect on shape and texture in terms of L_2-norm. However, individual characteristics are no longer retained (see Figure 3.7 (bottom row)).

3.2.3 Constraining Attributes

Since domains of attributes are not exclusive, there exist correlations between attributes. Previous systems for face modeling have relied either on morphing between existing example faces, or on additive changes that add multiples $r \in \mathbb{R}$ of vectors $\mathbf{a}$: $\mathbf{x} \mapsto \mathbf{x} + r\mathbf{a}$. Since attributes are correlated, however, these methods are suboptimal, as demonstrated by the following example: gender is correlated with the distance between eyes and eyebrows (Figure 3.8 (top right)), and if the user first selects a value for masculine appearance and then lifts the eyebrows, the result will look less masculine than desired, see Figure 3.8 (bottom left). In order to avoid iterative refinements, we introduced attribute constraints. If the user chooses to constrain gender, the masculine appearance will be retained by restricting subsequent modifications to the residual subspace of shapes and textures. Higher eyebrows are compensated automatically by a more male overall shape as shown in Figure 3.8 (bottom right).
In order to constrain attributes A_i, $i = 1, \ldots, m$, the current attribute values $c_{\mathbf{s},A_i}$ (shape) and $c_{\mathbf{t},A_i}$ (texture) are stored in vectors $\mathbf{c}_{\mathbf{s},\mathbf{A}}$ and $\mathbf{c}_{\mathbf{t},\mathbf{A}}$, respectively. For face shape vector $\mathbf{s}$ and texture vector $\mathbf{t}$, they are computed as

$$\begin{aligned} \mathbf{c}_{\mathbf{s},A_i} &= \langle\Delta\mathbf{s}, \Delta\mathbf{s}_{A_i}\rangle_{\mathbf{M}} \\ \mathbf{c}_{\mathbf{t},A_i} &= \langle\Delta\mathbf{t}, \Delta\mathbf{t}_{A_i}\rangle_{\mathbf{M}}\ . \end{aligned} \tag{3.5}$$

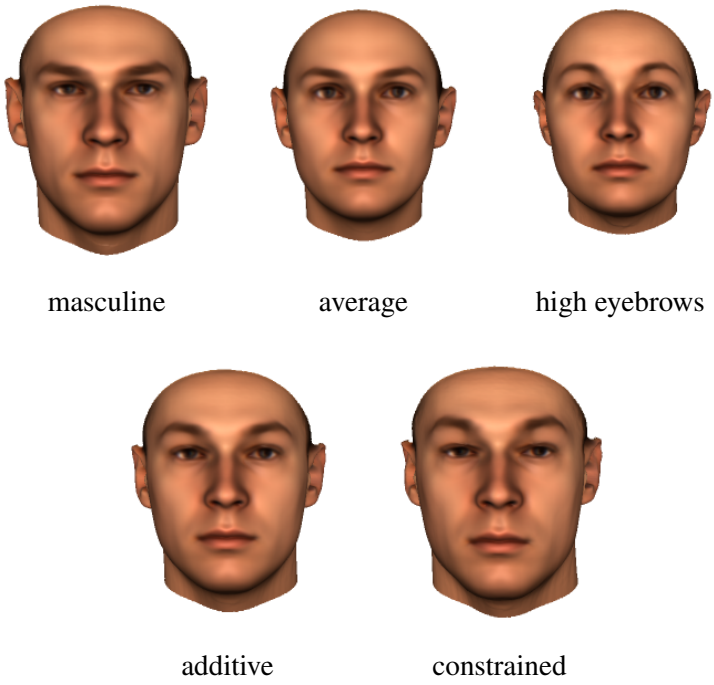


Figure 3.8: **Effect of constrained combination.** Top row: the shape attributes "masculine" and "high eyebrows" have partly conflicting effects on the average face. The masculine face has low eyebrows, while the face with high eyebrows appears feminine. Unlike the additive solution, where the attributes partly cancel out (bottom left), the constrained combination retains the selected masculine appearance and eyebrow distance (bottom right).

After modifications to the face, the face's shape and texture vectors must be updated to fulfill the constraints. A constrained attribute A_i is re-adjusted to the values $\mathbf{c}_{\mathbf{s},A_i}$ and $\mathbf{c}_{\mathbf{t},A_i}$ by adding multiples μ_j and ν_j to the current shape and texture vectors:

$$\begin{aligned} \Delta\mathbf{s}_{\text{new}} &= \Delta\mathbf{s} + \sum_{j=1}^{m} \mu_j \Delta\mathbf{s}_{A_j} \\ \Delta\mathbf{t}_{\text{new}} &= \Delta\mathbf{t} + \sum_{j=1}^{m} \nu_j \Delta\mathbf{t}_{A_j} \ . \end{aligned} \tag{3.6}$$

Thus, the set of constraints becomes

$$\begin{aligned} \mathbf{c}_{\mathbf{s},A_i} &= \langle \Delta\mathbf{s} + \sum_{j=1}^{m} \mu_j \Delta\mathbf{s}_{A_j}, \Delta\mathbf{s}_{A_i} \rangle_{\mathbf{M}} \\ &= \langle \Delta\mathbf{s}, \Delta\mathbf{s}_{A_i} \rangle_{\mathbf{M}} + \sum_{j=1}^{m} \mu_j \langle \Delta\mathbf{s}_{A_j}, \Delta\mathbf{s}_{A_i} \rangle_{\mathbf{M}} \\ \mathbf{c}_{\mathbf{t},A_i} &= \langle \Delta\mathbf{t} + \sum_{j=1}^{m} \nu_j \Delta\mathbf{t}_{A_j}, \Delta\mathbf{t}_{A_i} \rangle_{\mathbf{M}} \\ &= \langle \Delta\mathbf{t}, \Delta\mathbf{t}_{A_i} \rangle \mathbf{M} + \sum_{j=1}^{m} \nu_j \langle \Delta\mathbf{t}_{A_j}, \Delta\mathbf{t}_{A_i} \rangle_{\mathbf{M}} \ . \end{aligned}$$

photograph reconstruction scan

Figure 3.9: **Illumination-corrected texture extraction.** Left: photograph. Center: 3D reconstruction from the photograph on the left using illumination-corrected texture extraction. Right: scan of the same person for comparison.

The values for $(\mu_j)_{j=1}^m$ and $(\nu_j)_{j=1}^m$ are obtained by solving this system of equations. Now we can compute the output face using Equations 3.6. Operations are thus restricted to the subspace where all faces fulfill the constraints.

Note that for computing scalar products such as $\langle \Delta\mathbf{s}, \Delta\mathbf{s}_{A_i} \rangle_{\mathrm{M}}$ an explicit computation of the high-dimensional matrices ${\mathbf{C_s}}^{-1}$ and ${\mathbf{C_t}}^{-1}$ is not required, due to the decomposition of $\mathbf{C_s}$ and $\mathbf{C_t}$ into lower dimensional matrices obtained from PCA:

$$\begin{aligned} \langle \Delta\mathbf{s}, \Delta\mathbf{s}_{A_i} \rangle_M &= \langle \Delta\mathbf{s}, {\mathbf{C_s}}^{-1} \Delta\mathbf{s}_{A_i} \rangle \\ &= \langle \Delta\mathbf{s}, \mathbf{U}\mathbf{W}^{-2}\mathbf{U}^{\mathrm{T}} \Delta\mathbf{s}_{A_i} \rangle \\ &= \langle \mathbf{W}^{-1}\mathbf{U}^{\mathrm{T}} \Delta\mathbf{s}, \mathbf{W}^{-1}\mathbf{U}^{\mathrm{T}} \Delta\mathbf{s}_{A_i} \rangle \ , \end{aligned} \tag{3.7}$$

and analogously for ${\mathbf{C_t}}^{-1}$.

3.2.4 Generating New 3D Face Models from Images

Using an analysis-by-synthesis approach, the morphable model can be fitted to a facial image [BV99], see Figure 3.9 (left, center). The model parameters α_i and β_i, $i = 1, \ldots, N-1$, are refined along with rendering parameters such as camera position, image plane position and rotation, and intensity of light, until the difference between the face in the original image and a rendering of the textured 3D face model is sufficiently small. The process starts with the average face and an estimate of the rendering parameters by the user.
Since the morphable model cannot reconstruct texture details that were not present in the example faces, such as moles or scars, texture adaptation is performed subsequent to the model fitting process. After removing shading effects and shadows using the estimated rendering parameters, the error between the input image and the prediction by the model is minimized by adjusting the r,g,b values of the derived model's texture. Figure 3.9 gives a side-by-side comparison of the 3D scan of a head and of the reconstructed model using illumination-corrected texture extraction.
Since both model and rendering parameters are known, the extracted face model can be modified and rendered back into the original image.
In a similar manner, new 3D scans from faces can be expressed by the model.

source image target image composite

Figure 3.10: **Replacing a face in an image.** The face in the center image was replaced by the face from the left photograph to yield the new image on the right.

3.2.5 Replacing Faces in Images

In [BSVS04], Blanz et al. use their model in a semi-automatic technique to exchange a face in an image for a different face from another image with completely different pose, lighting conditions, and viewpoint. For this, the user must manually segment any foreground hair, such as fringes or sideburns, and click about seven feature points both on the images and on the average face. Then morphable model and rendering parameters are fitted to both images automatically. From the source image, the 3D model is taken, and from the target image, the pose and rendering parameters. The target person's hair style is also kept.

As starting values, the reconstruction algorithm takes the average face in front view and in the image center, with the light also coming from the front. The feature points are used to initialize the optimization process by considering the squared error between the feature points on the image and the positions of the feature points predicted from the model. The error function is chosen in such a way as to maximize the probability of rendering parameters and of the output face with respect to the variations in face space. The model is then fitted to the images as in the previous section and texture adaptation is performed. Where no texture can be extracted from the image, the estimated texture from the model is used, and boundaries are smoothed. Facial symmetry is exploited by mirroring the extracted texture from one side of the face, if the other side is occluded in the image.
Now the face extracted from the source image can be rendered into the target image with this image's rendering parameters. Due to the fact that both sets of parameters have been extracted based on the same morphable model, this works without any problems.
The final image is composited from target background, rendered 3D source model, and target foreground hair. In the example in Figure 3.10, the right image was obtained by replacing the face in the center image with the face from the photograph on the left.

4

Non-verbal Facial Animation

The naturalness of a talking head depends on a considerable number of factors related to the proper integration of visual and audio channel, i.e. of (visible) facial animation and (audible) speech. One important point is the generation of adequate lip movement synchronized to speech. Speech-related non-verbal facial expressions, such as raised eyebrows or blinks related to the structure of the spoken utterance, constitute another component. If such non-verbal facial expressions are not present in a talking head, it is perceived as soulless, highly artificial, and plainly boring. In addition, it has been shown that these speech-accompanying movements enhance understanding [HBG01], for instance by underlining important words or sentences and reinforcing utterance structure. A third important factor for naturalness in a talking head is the expression of emotions [ADMH04]. In human conversation, expressions of subtle and mixed emotions can be observed frequently, but the repertoires of most facial animation systems only include the six universal expressions joy, fear, anger, disgust, surprise and sadness. This is in part due to the limited visual data on intermediate emotions, and to the fact that they are not discrete, but rather continuous. Tsapatsoulis et al. [TRK$^+$02] therefore proposed to derive facial expressions for mixed emotions from a small number of fundamental emotions with known expressions.

The first part (Sections 4.1 through 4.3) of this chapter deals with our implementation of non-verbal speech-related facial expressions, both for the case of audio input (Section 4.2) and for the case of text input (Section 4.3). Section 4.1 gives a general introduction to the topic. In the second part (Section 4.4), an algorithm to create facial expressions for a continuum of intermediate emotions of varying intensity is introduced.

4.1 Non-verbal Speech-related Facial Animation

While listening to another closely visible person, for instance in a dialogue or movie close-up, the main visual focus of the listener is on the mouth and the eyes of the speaker. Facial expressions during speech, however, are not restricted to lip movement for sound production, but may include eyebrow raising, nose wrinkling, head movement, eye blinks, and more. These non-verbal facial expressions play an important role during speech. They enhance understanding by emphasizing words and syllables of special importance. Speech synchronization for animated characters should thus not be restricted to mouth movements, but should rather include other speech-related facial expressions as well in order to render the animations more vivid and believable.
Speech-related facial expressions are tightly coupled to the prosody of the utterance. Prosody, in turn, can be determined from the pitch of the signal. At the end of a question, for example, the

eyebrows raise in synchrony with the pitch. Accented words and syllables are also characterized by raised pitch.
Paralinguistic and psychologic research provide valuable insights for the automatic generation of non-verbal facial expressions related to speech: the close relationship between prosodic parameters of the speech signal and sentence structure on one hand and facial expressions on the other hand is described in the following. For an overview of the relevant literature see Section 2.1.2.

4.1.1 Psychological and Paralinguistic Background

Speech-related non-verbal facial expressions perform a variety of tasks. The different types can be categorized according to their function as follows:

- emblems
- illustrators
- punctuators
- regulators
- manipulators
- affect displays.

In the following, each group is addressed individually.

Emblems. Facial expressions that can replace speech are called *emblems*. English emblems are, for example, nodding for "yes", shaking one's head for "no", clicking one's tongue to express disapproval. Since they require some kind of semantic knowledge, we do not consider them in our implementation.

Illustrators. Facial expressions belonging to this category serve to underline important parts of speech, i.e. the speaker signals to the listener to pay special attention to the so-marked segments. Facial movements employed as illustrators are manifold. Most common are raising or lowering the eyebrows, moving the head, or nose wrinkling [Ekm79]. Other possibilities are tightening or widening of the eyes [Cho91] or eyeblinks. Which type is preferred varies between individuals. Several experiments (e.g. [CGB+96, HBG01]) prove that illustrators are closely linked to prosody.
Pitch related movement is the same for statements and questions. The only difference between questions and normal speech is that during questions the gaze of the speaker is directed towards the listener most of the time and always at the end [Cos91], while for statements the speaker does not constantly look at the listener.

Punctuators. *Punctuators* occur at the same positions as punctuation marks in written text, and they also perform the same task: to structure the utterance. An example are eye blinks during grammatical pauses or eyebrow movement [Ekm79].

Regulators. These turn-taking related facial expressions regulate interaction during conversations [Dun74, DF77]. At the beginning of his turn, the speaker emits a *speaker-state* signal: he turns his head away from the listener and starts gesticulation. Regulators further control the flow of speech by prompting the listener to take the turn (*speaker turn signal*) or by indicating that

the speaker wishes to keep his turn (*speaker within-turn*, *speaker-continuation*). If the speaker turns over the floor to the listener, he indicates this by several clues such as intonation, verbal expressions (e.g. "you know"), or termination of gesticulation. *Speaker within-turn* signals occur during grammatical pauses and involve the speaker looking at the listener to check whether the latter is still with him. They are often followed by *speaker continuation* signals (head shift away from listener), especially if an *auditor back-channel* occurs before the end of a unit of analysis. *Back-channel* signals by the listener include brief verbalization such as "m-hm" or "a-ha", sentence completions, requests for clarifications, brief restatements, and head nods and shakes. They indicate to the speaker that the addressee is still following.

Manipulators. *Manipulators* are movements that accommodate some physical need, such as licking one's lips or eye blinking to moisten lips or eyes, respectively. They are performed unconsciously.

Affect Displays. Facial displays of emotions can either be genuine or merely a reference to an emotion felt during an event that now constitutes the topic of conversation, or they can serve as some kind of emblem. Instead of expressing disapproval verbally, facially expressing disgust without actually feeling this emotion will get the message across equally well. Affect displays share the problem of emblems that semantical and context knowledge is indispensable for inserting them into animations in a meaningful way. Since semantical analysis is out of the scope of our work, we solve this problem for the case of text input by allowing the user to insert emoticons into the input text. The corresponding expressions are displayed at the indicated positions by the talking head. This approach is described in Section 4.3.

We implemented illustrators, punctuators, regulators and manipulators for two different input situations. Section 4.2 explains how position and duration of non-verbal speech-related facial expressions can be deduced from the information present in the audio signal, whereas Section 4.3 deals with the case of text input. The text is processed by a coupled text-to-speech system, which performs linguistic analysis of the input. From these intermediate results, both audio signal and facial animation are generated.

4.2 Non-verbal Speech-related Facial Animation from Audio

As mentioned above, prosodic parameters of the speech signal such as slope and range of the fundamental frequency $\mathbf{F0}$ are coupled to facial expressions [CGB+96]. By extracting these prosodic parameters from the speech signal, we are able to automatically generate facial expressions that match the prosody of the utterance [AHS02a].

This section describes a method to compute the following non-verbal facial expression from speech automatically:

- head and eyebrow raising and lowering in accordance to the pitch
- gaze direction, movement of eyelids and eyebrows, and frowning during thinking and word search pauses
- eye blinks and lip moistening as punctuators and manipulators
- random eye movement during normal speech.

The intensity of facial expressions is additionally controlled by the power spectrum of the speech signal, which corresponds to loudness and intensity of the utterance.

4.2.1 Generating Non-verbal Facial Expressions

We have implemented our method for automatic generation of non-verbal facial expressions from speech as a module in the MEDUSA facial animation system, cf. Section 3.1. The generation of both speech synchronized mouth movements and non-verbal facial expressions is carried out in a preprocessing step, which takes about seven seconds (5 s for analysis of the speech signal and 2 s for animation generation) for a speech signal of 30 seconds duration on a Pentium 4 1.7 GHz dual processor PC. Once the speech synchronized animation parameters have been generated, the animation runs in real-time (40 fps for animation, ≈100 fps for rendering).

Facial Expressions from Pitch

To automatically generate head and eyebrow movement from the speech signal, we first extract the pitch values of the utterance at a sampling distance of 10 ms. We use the Snack Sound Toolkit [Sjö01], which provides a variety of routines for speech analysis.
Since the production of unvoiced phonemes such as /p/ or /f/ does not involve vocal chord vibration, the notion of pitch does not exist for these sounds. Hence the pitch value is zero, which leads to a very rugged appearance of the pitch curve. Therefore we eliminate these zero values and approximate the remaining pitch values using a B-spline curve. Next, the local minima and maxima of this curve are determined. Their positions and values, however, do not correspond exactly to the minima and maxima of the original pitch curve. Thus, for every local maximum of the B-spline curve, position and value of the maximum of the original pitch data from the interval between the preceding and succeeding turning point of the B-spline curve are retrieved. An analogue process is performed for the local minima. The "reconstructed" original extrema are then used for the generation of head and eyebrow movement.
For each of the so determined maxima it is decided whether the differences between its value and the values of the preceding and succeeding minima exceed a given threshold. This threshold is to a certain degree speaker dependent: some people use greater amplitude of voice melody than others. For those maxima where the threshold is exceeded, the head is raised. The amount of head movement depends on the magnitude of the maximum. We typically generate head movements of at most three degrees rotation about the horizontal axis.
For each minimum, the difference values to the preceding and succeeding maxima are computed. Again, if the differences are larger than a given threshold, head movement is generated. In this case, the head is rotated back into its neutral position. This combination of upward and downward movement of the head supports accentuation of the speech. Figure 4.1 depicts different stages of the processing of the speech signal and the resulting animation parameters.
Both raising and lowering of the head are synchronized at the phoneme level, i.e. they are realized at the phoneme boundary closest to the computed point of occurrence.
It is necessary to use only the most prominent maxima and minima, because otherwise too much head movement would be generated. In order to avoid monotonous movement, the head is also randomly turned or tilted slightly to one side from time to time.
Head movement is often accompanied by analogue eyebrow movement: eyebrows are raised for high pitch and lowered again following intonation. In our approach, eyebrow movement is

generated using the same method as for the head rotations. Figure 4.1 (bottom) gives an example of pitch dependent eyebrow raising.
According to Cavé et al. [CGB+96], only the magnitude of the left eyebrow's movement is related to the **F0** pattern. This is taken into account by varying the degree of eyebrow raising for the left side according to the value of the current maximum, while the right part of the *occipitofrontalis* is always contracted by the same amount. The duration of eyebrow raising is not correlated to the magnitude of the movement. This is inherently included in our implementation, since the duration depends only on the time step between the previous minimum and the maximum.

Thinking and Word Search

During prolonged or filled pauses (e.g. "... errr ...") in a monologue, the speaker is typically either thinking about what to say next or searching for words. In both cases, similar facial expressions are exhibited: the gaze is directed at an immobile, fixed location to reduce visual input [Ekm79]. This location is usually either somewhere on the floor or up in the air. When people look up, they also raise their eyebrows. One possible explanation for this is an increase in the field of view when the eyebrows don't occlude part of the vision [Ekm79]. On the other hand, when people look at the floor while searching for answers, they often show a slight frown. We have implemented this word search and thinking behavior during pauses (see Figure 4.2). The duration of pauses that justify thinking and word search behavior seems to be speaker dependent and can hence be adjusted by a parameter. We use a probability of 25 % for the talking head's showing a frown during a thinking pause. In all other cases, both gaze and eyebrows are raised.

Punctuators and Manipulators

As already mentioned, punctuators are facial expressions that are used during speech at the same positions as punctuation marks in written text, thereby helping to structure the flow of speech. A good example for such punctuators are eye blinks. In our implementation, we generate eye blinks at the beginning of pauses.
Since eye blinks also serve the physical need to keep the cornea moist, they fall into the category of manipulators as well. Such blinks occur on average every 4.8 seconds [PBS96]. Hence, if the time elapsed between the previous and the next blink exceeds the threshold of 4.8 seconds, an additional blink is inserted. These involuntary eye blinks have considerable impact on the lifelikeness of the character.
As described by Pelachaud et al. [PBS96], eye blinks consist of a closing interval (average duration: 1/8 s), the apex (average duration: 1/24 s), during which the eyes remain closed, and an opening interval (average duration: 1/12 s), where the eyes open again. Eye blinks are also synchronized to the speech: beginning of the closing, the apex, and the opening coincides with the nearest phoneme boundaries. This behavior is simulated in our implementation.
Besides involuntary eye blinks, another example for a manipulator is the moistening of the lips during extended speech periods. This can be implemented by letting the synthetic character lick its lips during pauses that match the average lip moistening frequency best. During pauses where a thinking or word search expression is exhibited, the tongue motion should be slower, because the speaker is concentrating entirely on what to say next.

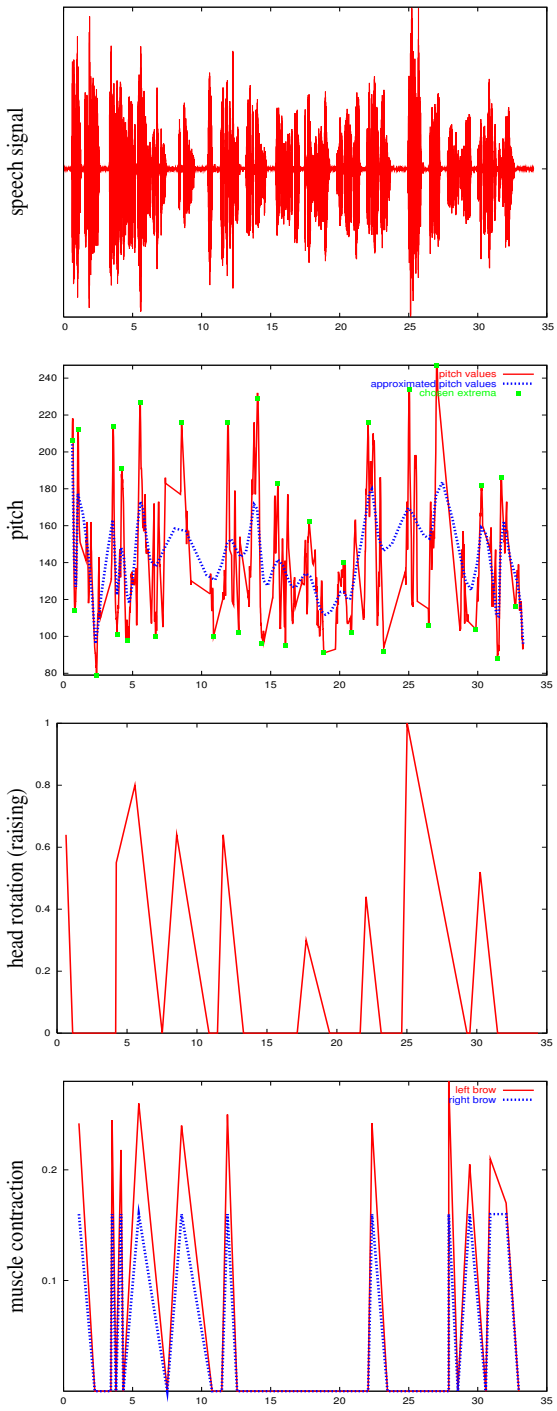


Figure 4.1: **Processing the speech signal.** In all diagrams, the x-axis shows time [sec]. Top to bottom: waveform of input speech signal (about 33 s); original pitch values [Hz] (red) and corresponding B-spline curve (blue) with maxima (green squares); resulting head movement; muscle contractions of the *frontalis* muscles, which are responsible for eyebrow movement (red: left brow, blue: right brow). Note that the movement of the left brow is scaled according to the magnitude of the pitch value.

Figure 4.2: **Snapshot of a reflective moment during speech synchronized facial animation.** Left: only mouth movement is generated from the speech signal. Right: additional movement of head, eyes, and eyebrows is generated automatically from prosodic parameters. The character looks up, raises both head and eyebrows, and slightly tilts its head.

Random Eye Movement

During normal conversation, the speaker does not always look at the listener [Cos91]. Moreover, eyes are almost constantly in motion. For an animated character lacking this behavior, the gaze is staring and dead. We have included additional random eye movement into our facial animations. Here it is important that the eye positions do not differ too much between consecutive movements. Otherwise the movement seems erratic and the character might appear agitated. As with all upward and downward eye movements, it is crucial that the lids accompany the eyeballs: if a person's gaze is directed downwards, the eyelids also close to a certain degree. Contrariwise, if one looks up, the eyelids open more to prevent an occlusion of the field of view.

Volume-controlled Intensity

Loudness primarily influences the magnitude of speech-related mouth movement for vowels. Additionally, it is also a good indicator for the distance of the person we are talking to. If somebody wants to pass on information to a person standing several meters away, he must speak louder in order to be understood. For the same reason he may also choose to intensify his speech-accompanying facial expressions. A very slight head movement, for example, is not perceivable at greater distances, so the speaker may want to nod more vigorously. Therefore we do not only scale lip movement by the power of the signal, but allow this for pitch related facial expressions as well. The extent to which they are scaled can be regulated by a parameter. This allows us to model differences in the behavior of the animated characters.
Using the Snack sound toolkit [Sjö01], we extract a windowed power spectrum of the speech signal and fit an approximating B-spline curve to it. An interpolating polynomial is fitted to the local maxima of this B-spline curve and normalized to a $[0, 1]$ range. It indicates the relative loudness of the speech signal. These relative loudness values are individually weighted for each animation parameter and used to scale the intensity of facial expressions. The weight for jaw rotation, for instance, is greater than the weight for eyebrow movement. The weights can be modified to model characters with different attitudes. Figure 4.3 shows the windowed power spectrum for an example sentence together with the approximating B-spline curve and its maxima.

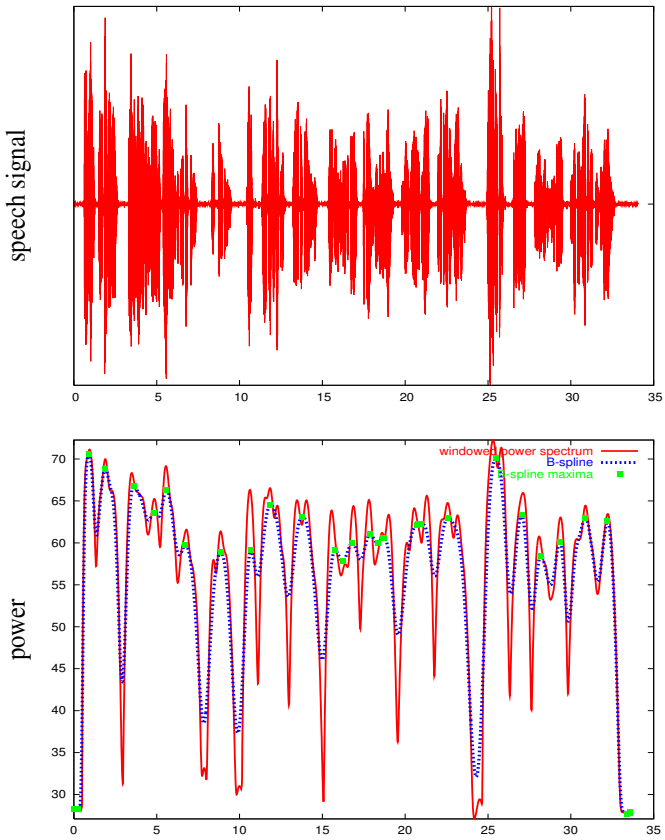


Figure 4.3: **The windowed power spectrum.** Top: waveform of input speech signal. Bottom: windowed power spectrum [dB] (red), approximating B-spline curve (blue), and maxima of the B-spline (green squares). Again, the x-axis indicates time. This information is used for scaling facial expressions with respect to loudness.

4.2.2 Results

Incorporating non-verbal speech-related facial expressions into our facial animations definitely improved their naturalness and made them more appealing. Although the movements are generated by rules, random variations are taken into account to prevent the facial expressions from being entirely predictable. Some predictability, however, should remain indeed, since the accentuating facial expressions of humans tend to be predictable as well.
By specifying weights and frequencies for the movements of head, eyes, and eyebrows, different synthetic characters can be designed that exhibit different ways of visually accentuating their speech. This is also the case for real humans: some people habitually underline important parts of their utterances by eyebrow movement, and some by nodding. The frequency and amplitude of such movements depend highly on the temperament and culture of the individual as well. We would expect an Italian, for instance, to show much more facial and body gestures than a person from Northern Europe.
Figure 4.4 shows several snapshots from a facial animation sequence synchronized to a speech signal both with and without additional non-verbal facial expressions. The animation that includes non-verbal facial expressions clearly looks more convincing and lifelike.

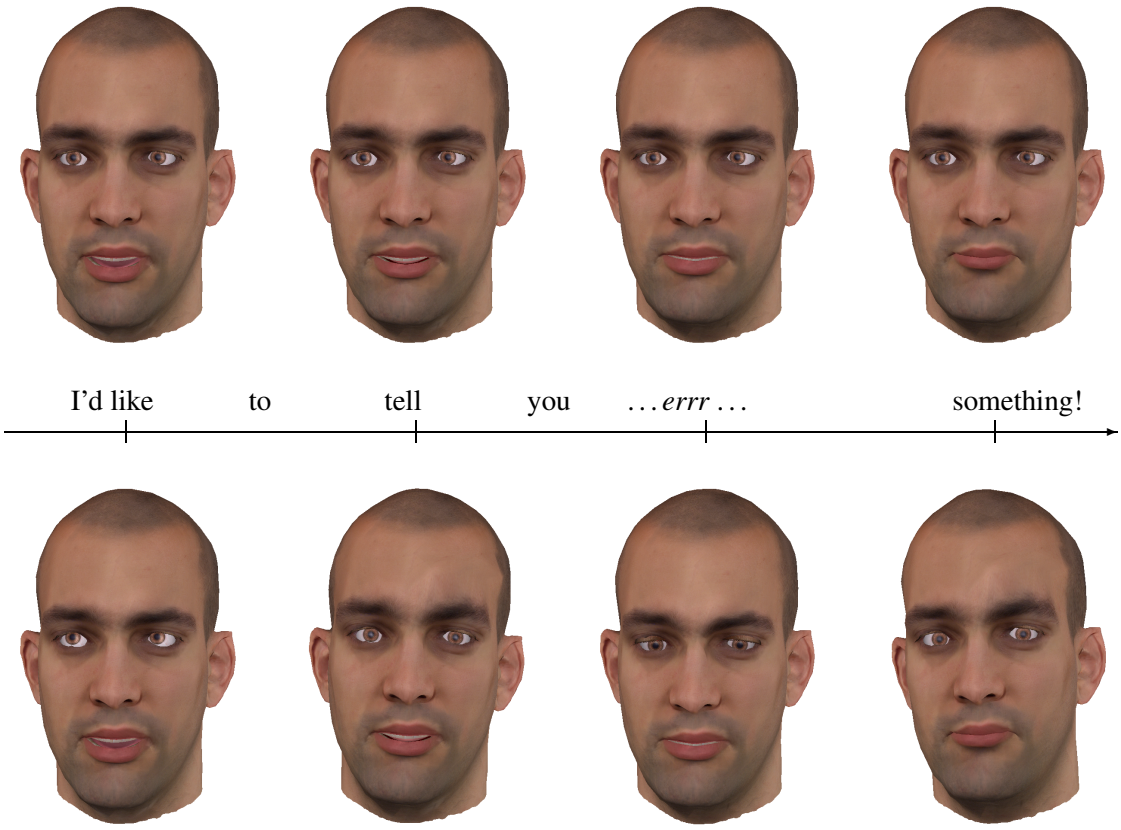


Figure 4.4: **Snapshots of facial animation from audio.** This facial animation sequence was synchronized to a speech signal with the textual representation: "I'd like to tell you ...*errr* ...something!". Top row: movements of lips and jaw are generated from the speech signal. Bottom row: additional non-verbal facial expressions are created automatically from a paralinguistic analysis of the speech signal.

4.2.3 Conclusions

We have presented a method to automatically generate non-verbal facial expressions from a speech signal. In particular, our approach addresses the movement of head, eyes, eyelids, and eyebrows depending on prosodic parameters such as pitch, length and frequency of pauses, and the power spectrum of the input signal. These parameters are extracted automatically from the speech signal and control our facial animation parameters in accordance to results from paralinguistic research. Resulting animations are definitely more natural and vivid compared to speech synchronized animations that control mouth movements only.
Integrating a statistical model for gaze during speech as described in [LBB02] would make gaze behavior much more lifelike by integrating saccades, i.e. the somewhat jerky way eyes move.
Incorporating facial expressions that match the emotion conveyed by the speech signal would enhance the realism of our system considerably. A learning based approach as used by Chuang et al. [CDB02] to extract the emotions neutral, happy and angry, and by Cao et al. [CTFP05] for the emotions neutral, sad, angry, happy and frustrated allows classification of a limited number of emotions from the speech signal.

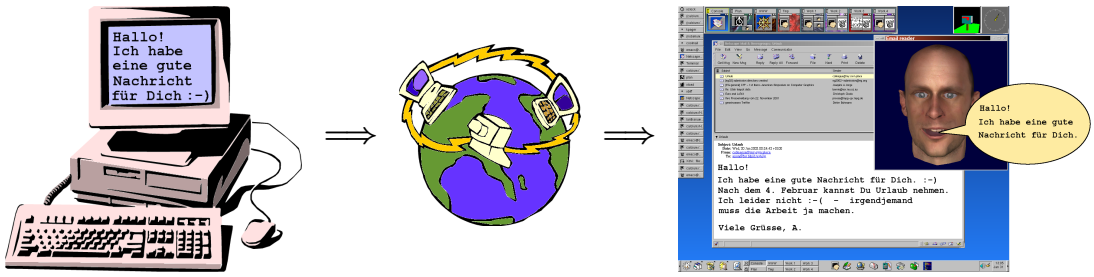


Figure 4.5: **Possible application scenario.** A text message is embellished with additional emoticons, sent over the network as an e-mail, and read on the receiver's computer by a virtual character showing corresponding emotions.

The following section approaches the problem of emotional displays during speech for the case of text input by allowing the user to insert emoticons into the text.

4.3 Non-verbal Speech-related Facial Animation from Text

Synthetic audible speech from text input has improved a lot in recent years by taking into account intonation and syntactic importance of individual words and phrases. The linguistic analysis of text input, which is necessary to perform speech synthesis, can be used to additionally drive a facial animation system that generates speech synchronized mouth movements as well as non-verbal speech-related facial expressions [AHK+02]. Additionally including emoticons in the text input allows to display emotions, which can be useful in applications such as the one depicted in Figure 4.5.

As a step into the direction of fully automated facial animation from text, we coupled the MEDUSA facial animation system (Section 3.1) and the MARY text-to-speech system (Section 4.3.2). MARY has the advantage of creating additional output suitable for generating facial expressions. This enabled us to implement rules to automatically add synchronized non-verbal speech-related facial expressions to our lip sync animations.
This section is organized as follows: a brief introduction to text-to-audible-speech synthesis is given in Section 4.3.1, followed by an overview of the MARY text-to-speech system in Section 4.3.2. Section 4.3.3 motivates our approach by explaining why MARY and MEDUSA make such a good team for bimodal synthetic speech production. A sketch of the combined system can be found in Section 4.3.4. Section 4.3.5 describes how speech-related facial animations are generated from the output of the text-to-speech system, with a detailed example given in Section 4.3.6.

4.3.1 Text-to-Speech Synthesis

Text-to-speech (TTS) synthesis [Dut97] is a method for converting written text into audible speech. It consists of a text analysis part, generating a symbolic representation of a spoken utterance including a phonetic transcription of the words, followed by the actual speech synthesis part, in which the symbolic representation is converted into audible speech.
Early systems, such as the MITalk system [AHK87], employed formant synthesis algorithms leading to relatively unnatural, "robot-like" voices. A major improvement in naturalness was

brought about by concatenative synthesis techniques [DPP+96, BC95], which produce synthetic speech by re-sequencing human recorded speech samples. These new synthesis techniques have increased the intelligibility of synthetic speech considerably. Naturalness, however, is still a prime issue. The expressive capabilities of synthetic voices were augmented by modeling vocal emotions [Sch01, Sch04a, Sch04b]. These systems are used, for instance, in audio-visual speech synthesis [Sta00].
Speech synthesis systems that are to be used in conjunction with facial animation need to provide intermediate processing results such as timing information in addition to the resulting speech. The most wide-spread research system for speech synthesis, the open-source FESTIVAL system [BTC99], uses its own, relations-based data representation for this purpose. New systems using XML-based internal data representations, such as BOSS [KSV+01] and MARY [ST03, Sch04b], make the output of partial processing results a straightforward task. The XML data can be further analyzed by subsequent processing components using standard XML parsers.

4.3.2 The MARY Text-to-Speech System

Our system uses text-to-speech synthesis techniques for the creation of the speech signal as well as for the description of the speech signal structure needed for the audiovisual synchronization. The MARY TTS system for German (and lately also English) [ST03, Sch04b, Sch05] was integrated into our system for performing these functions.
MARY creates speech from text in five major processing steps. In a first step, a shallow linguistic analysis of the plain text input is performed, using statistical algorithms trained on large text corpora [Bra00]. This analysis component consists of a tokenizer identifying word and sentence boundaries including, in particular, the role of dots (abbreviation, ordinal number, or sentence-final), a part of speech ("noun", "adjective") tagger, and a local syntactic parser using statistically trained trigram models [SB98]. In the case of special text types such as poems, the tokenizer performs an additional segmentation at line breaks, needed for the line-based speech rhythm typical for poem reading. In a second step, a phonetic transcription is assigned to each word. This component performs a lexicon lookup for each of the words, and assigns the phonetic transcription to the known words. Unknown words, such as proper names, are transcribed by means of a set of transcription rules. Thirdly, an intonation contour including accents and boundaries is assigned to each sentence on the basis of the linguistic analysis. Accents are placed on content words, leading to a pitch excursion and thus to a perceptual prominence needed by listeners for understanding the meaning of the text. Intonation rises and falls at boundaries to reflect the sentence type (question vs. statement). In a fourth step, the symbolic information about phonetic transcription and intonation is used for determining the precise acoustic parameters of the utterance, each phoneme's duration (in milliseconds) and the shape of the intonation contour (using a sequence of target points with fundamental frequency expressed in Hertz). In a final step, these values are interpreted by a waveform synthesis algorithm to create a speech signal.

The MARY system is particularly well suited for integration into our facial animation system. It was designed to provide, in addition to the synthesized speech, as much explicit information as possible about the individual processing steps it runs through. This information is valuable for a number of reasons. Most obviously, exact timing information of the individual sound segments produced (determined at the second and the fourth processing step described above) is needed for a proper synchronization of lip movement with the sound. In addition to that, higher level

emoticon	emotion	emoticon	emotion
:-)	happy	;-)	kidding
:-(	sad	>:-<	angry
:-o	surprised	:-|	disgusted

Table 4.1: **Emoticons.** These emoticons are available in our system to manually add facial expressions of emotions to animations of speech.

information is available, such as the type (step three) and timing (step four) of accents, which correspond to the important bits of the sentence. Proper analysis of these accents allows the rendering of appropriate time-aligned facial gestures, thus conveying a truly multi-modal pattern of accentuation expression, contributing to both naturalness and intelligibility of the synthesized audio-visual speech [CPB+94]. A further type of high-level information that can be extracted from the MARY output is the type and location of boundaries (pauses), including a differentiation between sentence-internal and sentence-final pauses, as well as sentence type (determined in step one, and specified in steps three and four). The ability to make such distinctions is a prerequisite for assigning proper non-verbal facial expression, e.g. gaze behavior which differs between questions and statements, and between sentence-internal and sentence-final pauses.
A significant advantage of the MARY TTS system is that its data representation is based on XML. Among other things, this allows XML-based markup to be provided in the text input and to be passed on to subsequent processing components such as, in this case, the facial animation component, see also the example in Section 4.3.6. This property has been put to use for the automatic expression of emotions in the speech-synchronized facial animation.
As a simple means for the textual representation of emotions, so-called *emoticons* ("smileys", "frownies", etc.) are widely used, particularly in e-mails. We propose a simple but effective method for interpreting these emoticons for generating appropriate facial expressions. As a first list, the emoticons from Table 4.1 are recognized by our system.
These emoticons are automatically translated into XML-based emotion markup before the text is fed into the TTS system. When the research reported in this chapter was conducted, the MARY system merely passed this information on to the visual generation component. The implementation of appropriate vocal changes, reflecting the emotion in the synthetic voice, was only completed after the end of this project (see Section 4.4.2).

4.3.3 Speech Animation and Synchronization

The data required for the synchronization of lip movement to audio is provided by the TTS system, which generates the corresponding phonemes along with their durations from plain text input in a SAMPA representation. SAMPA (*Speech Assessment Methods Phonetic Alphabet*) is a machine-readable phonetic alphabet. The algorithm proposed by Cohen and Massaro [CM93] that was implemented for lip sync in MEDUSA is well suited to handle time-based phonetic descriptions of speech such as the SAMPA representation.
It turned out that the muscle-based facial animation approach of MEDUSA is very well suited for automatically generating speech synchronized animation sequences. Since the virtual facial muscles are defined on a reference head and can be easily transferred to a new head model [KHYS02], no tedious parameter tuning is required for an individual head model: for each phoneme, the muscle contractions that have been found to result in the correct corresponding viseme on the reference head can simply be re-used. Finally, since audio signal and muscle

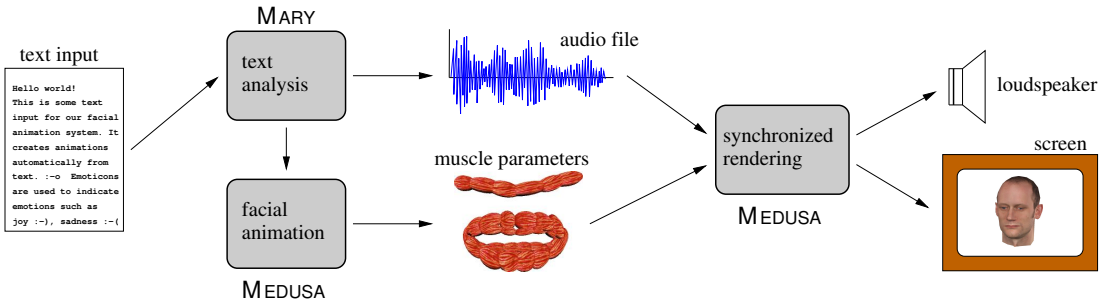


Figure 4.6: **System Components.** Text input is converted into a speech signal by the MARY TTS using linguistic analysis, which additionally drives the muscle-based facial animation system MEDUSA by providing information required for lip sync and non-verbal speech-related facial expressions. The emoticons in the text input are handed down by the TTS to MEDUSA, which inserts corresponding facial expressions into the animation. Rendering and audio output are synchronized in the final animation sequence.

contractions are generated from the same time-based data, the resulting animations exhibit perfect synchronization of audio and video (see Section 3.1.4).

4.3.4 System Overview

Our facial-animation-from-text system consists of three major components:

1. MARY as text analysis module
2. MEDUSA as facial animation module
3. a module to synchronize rendering and audio output.

Figure 4.6 illustrates the connections between these modules.

The text analysis module performs linguistic analysis of the input text and creates a synthesized speech signal using a male or female voice. This process is described in more detail in Section 4.3.2. The results of the linguistic analysis are passed on to the facial animation module, which performs two different tasks: from the phoneme-based representation of the input text, speech-synchronized muscle contraction values for the facial muscles used for speech are generated. This process takes into account coarticulation (cf. Section 3.1.3). Additional high-level linguistic information such as different types of accents, pauses, and sentences is converted into non-verbal speech-related facial expressions, which are also represented as muscle contraction parameters, see Section 4.3.5. Finally, the animation sequence resulting from the muscle contraction values is rendered in real-time (as described in Section 3.1.4), in synchrony with the audio output of the speech signal.

4.3.5 Generating Facial Expressions

The decoration of speech with non-verbal facial expressions in daily face to face communication is so habitual to us that we are not even consciously aware of it. If, however, these facial expressions are missing from a facial animation, we perceive the presentation as boring. It even

becomes more difficult to understand the meaning of the message due to the lack of visual structuring. As outlined in Section 4.1.1, apart from providing structure, non-verbal facial movement also serves to accentuate words to indicate high importance, to facilitate turn-taking between speaker and listener, to underline that a question is being asked, to express emotions or opinions, and also to fulfill physical needs such as cornea moistening.

Extraction of Speech-Related Facial Motion

Our animations include automatically generated speech-related eye movements, blinks, eyebrow gestures, and head movement. All movement is synchronized at the phoneme level.
In a conversation between two or more real human beings, a large amount of turn-taking occurs where the flow of the dialog is controlled using facial gestures. Since we have only one virtual character talking to the user, we model just a small subset of these turn-taking gestures to emphasize certain parts of the text spoken by the virtual head, for instance, facial expressions during questions.

Eye blinks. Eye blinks are created as punctuators during pauses to structure the utterance. Additional eye blinks are inserted to keep up with the natural rate of cornea moistening, which occurs on average once every 4.8 seconds [PBS96].

Questions. When posing a question, the eyes of the virtual character are directed at the user, who is assumed to sit directly in front of the screen. This kind of "making eye contact" increases the impression of the virtual face being aware of its vis-à-vis. Missing eye contact is unsettling, because the user feels unsure if it is really him who is being addressed by the question. Additionally, questions are marked by raising the eyebrows and the head on the last word, lasting over a potentially following pause. The TTS system provides information about word and sentence boundaries as well as on the type of the sentence, which ensures a correct placement of the corresponding facial behavior.

Expressions linked to intonation. We generate eyebrow and head movement from the intonation information provided by the TTS system. The intonation data is given in the form of `(time,value)` pairs, where the value indicates the fundamental frequency or pitch value at the given time. From this data, the local pitch extrema are extracted. In the animation, the head is raised at every local pitch maximum proportionally to the pitch value. For every local pitch minimum, the head returns to its rest position, independent from the actual pitch value at the minimum. To avoid repetitive movements, the head is randomly tilted sideways slightly in about 75 % of all head raises.
Eyebrow raising is implemented similarly, but occurs less often: head movement is more frequently used, because it is easier to detect over distances [HBG01]. According to Cavé et al. [CGB+96], we specify the probability of occurrence of an eyebrow gesture at a pitch maximum as 0.71. The randomized distribution of actual eyebrow gestures also makes the animation less predictable. Following the observations made by Cavé, only the amplitude of the left eyebrow movement depends on the pitch value. Thus the right eyebrow is always raised by a constant amount.
It is also important that the speaker does not direct his gaze away from the listener at intonation maxima. Otherwise conflicting information would be conveyed: looking away from the listener could indicate low importance, while intonational stress, head and eyebrow movement signal

Figure 4.7: **Snapshots of facial animation from text.** This facial animation sequence was generated from the text input "*Hi! ;-) Can you show me the way to the conference hall? :-)*". Top row: movements of lips and jaw are synchronized to the audio signal created from the text input. Bottom row: additional non-verbal facial expressions are generated from the results of a linguistic analysis of the text input.

that important information is communicated. In fact, the speaker may look explicitly towards the listener in order to emphasize what he is saying [AC76]. In our implementation, the talking head glances at the listener during 70 % of all accented syllables, and does not shift its gaze further away from the listener in all other cases.
The bottom row of Figure 4.7 shows an example for intonation-related facial expressions. The word *conference* is emphasized. Therefore, the fourth snapshot in the bottom row, which was taken during articulation of this word, shows the talking head with eyebrows and head raised, glancing at the listener.

Word search. Explicit eye movement is also performed during prolonged pauses within a sentence, especially when accompanied by an '*errr*' sound. In this case it can be assumed that the virtual character is searching for words. We simulate this behavior by letting the character either look down and frown (with a probability of 25 %), or raise its eyebrows and stare at an imaginary ceiling. The talking head also shifts its gaze slightly to the right: Andersen [And99] reports that rightward lateral eye movements are associated with verbal and linguistic activity.

Regulators. Argyle and Cook [AC76, p. 121] found that the speaker looks at the listener during grammatical breaks both as a signal and to obtain visual feedback. From the viewpoint of turn-taking, Duncan and Fiske [DF77] call this phenomenon a *speaker within-turn* signal. We have implemented this behavior through glances during 70 % of all pauses that coincide with intonational phrase boundaries (cf. the definition below).
At the beginning of its monologue, the talking head averts its gaze and turns away from the listener as a turn requesting signal (*speaker state*). At the end of the utterance, the character

looks at the listener to indicate that it has finished (*speaker turn* signal).

Gaze. If, apart from the explicit eye movements described above, the eyes of the talking head are kept completely still over the course of the animation, the awkward impression of a "dead stare" is evoked. Hence, we vary the view direction randomly within a small range, making the character appear more lively. Together with the eyeball rotation, the eyelids open or close slightly when looking up or down, respectively. The amount of eyelid opening depends also on the tilt angle of the head: if the head is tilted downward, the eyelids are opened more widely to allow for a straight viewing direction, and vice versa.

Facial Expressions from Emotion Tags

Emotions are embedded in the text input via emoticons and translated to XML markup included in the final rich XML representation generated by the text analysis component (cf. Section 4.3.2). We use this information in the animation module to generate appropriate emotional facial expressions. The first snapshot in the bottom row of Figure 4.7 shows the wink and the smile that accompany the friendly greeting indicated by the `;-)` symbol.
The apex of the emotion is determined by the position of the emoticon in the text input. We assume an intensity of 100 % of the corresponding emotion at its climax. The area of influence of an emotion is the *intonational phrase* during which it occurs. An intonational phrase is a natural unit in speech production, which often comprises only part of a sentence and is typically surrounded by pauses. Towards the borders of these phrases, the intensity of the emotion decreases linearly to zero. If an emotion is specified between two phrases, the emotion stretches over both phrases. For instance, in the poem in Table 4.2, the `:-(` emoticon placed after the colon in the second line of the second verse represents such a case. In our approach, emotional expressions and speech animation are combined by summing up the respective muscle contraction values.

4.3.6 An Example

At the time we conducted the research described in this section, the text-to-speech component supported the German language only. An extension to the English language is now available. Among other examples, we automatically generated a facial animation sequence from the poem "*Die zwei Parallelen*" by the German poet Christian Morgenstern (see Table 4.2). The emoticons `;-)`, `:-(`, `:-o`, and `:-)` in the poem have been inserted manually.
In the XML representation of this poem (see Table 4.3), the original text tokens are highlighted in grey, e.g. `gingen`. The animation module uses the original text representation only to find question marks. Sentences are bracketed by `<div>` and `</div>`. Text analysis and speech synthesis information pertaining to a single word is stored in a `<t ...> ... </t>` pair. The attribute `sampa` contains the MARY SAMPA phoneme representation of the word. This information is used to determine the temporal position of the word in the phoneme representation of the input text. In this way, non-verbal facial expressions derived from XML tags are synchronized to the speech-related mouth movements generated from the phoneme representation. The XML tags for the beginning and end of an intonational phrase are color coded as `<phrase>` and `</phrase>`, respectively. During the second phrase, the emotion `;-)` ("kidding") is specified as `<emotion type="kidding"/>`. The corresponding emotion is blended in at the beginning of "*ins*", reaches its maximum at the specified position, and is faded out at the end of "*hinaus*".

Es gingen zwei Parallelen
ins `;-)` *Endlose hinaus,*
zwei kerzengerade Seelen
und aus solidem Haus.

Sie wollten sich nicht `:-(` *schneiden*
bis an ihr seliges Grab: `:-(`
Das war nun einmal der beiden
geheimer Stolz und Stab.

Doch als sie zehn Lichtjahre
gewandert neben sich hin,
da wards dem einsamen Paare
nicht `:-o` *irdisch mehr zu Sinn.*

Warn sie noch Parallelen?
Sie wußtens selber nicht, –
sie flossen nur wie zwei Seelen
zusammen durch ewiges Licht.

Das ewige Licht durchdrang sie,
da wurden sie `:-o` *eins in ihm;*
die Ewigkeit verschlang sie
als wie zwei Seraphim. `:-)`

Table 4.2: **Example text.** "*Die zwei Parallelen*" by Ch. Morgenstern (1905).

4.3.7 Conclusions

Our system provides an easy-to-use method to generate facial animation from text input with optionally included emoticons. Due to this simple interface and the full automation of the process after specifying the text, applications such as depicted in Figure 4.5 can be supported easily. The facial expression cues improve the quality of speech animation significantly. Figure 4.7 shows a comparison of some snapshots from such a facial animation sequence. Processing is fast enough to offer interactive response times in a dialog setting. More work needs to be done, though, to improve the naturalness of conversation in a human-machine dialog: the system would need to have knowledge about the emotional state of the user to be able to show appropriate reactions.
For emotion expression to become more convincing, vocal emotion cues must be delivered in addition to the facial emotion expression. Emotional speech synthesis is possible now [Sch01, SCDC$^+$01, Sch04b], see Section 4.4.2. It is to be expected that the combined expression of emotion via both the visual and the auditory channel will improve the perceived naturalness. The following chapter addresses some of the questions arising in this context.

4.4 Speech and Emotion

Apart from lipsync and non-verbal speech-related facial expressions, a third important factor for naturalness in a talking head is the expression of emotions [ADMH04]. Unfortunately, the general "toolbox" for modeling emotions and hence also their facial or vocal expression is not yet very well developed. While attempts are under way to organize the vocabulary and models

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE maryxml SYSTEM "http://mary.dfki.de/MaryXML.dtd">
<maryxml>
<speaker gender="male">
<phonology nasal_assimilation="on" precision="precise"
    schwa_elision="on">
<div>
<phrase>
<t g2p_method="lexicon" pos="PPER" sampa="'?{s"
    syn_attach="1" syn_phrase="_">
Es
</t>
<t g2p_method="lexicon" pos="VVFIN" sampa="'gI-N@n"
    syn_attach="1" syn_phrase="_">
gingen
</t>
<t accent="l+h*" g2p_method="lexicon" pos="CARD"
    sampa="'tsvaI" syn_attach="1" syn_phrase="NP">
zwei
</t>
<t accent="l+h*" g2p_method="lexicon" pos="NN"
    sampa="pa-ra-'le:-l@n" syn_attach="0" syn_phrase="NP">
Parallelen
</t>
<boundary breakindex="4" tone="h-l%"/>
</phrase>
<phrase>
<t g2p_method="lexicon" pos="APPRART" sampa="'?Ins"
    syn_attach="1" syn_phrase="PP">
ins
</t>
<emotion type="kidding"/>
<t accent="l+h*" g2p_method="userdict" pos="ADJA"
    sampa="'?{nt-lo:z@" syn_attach="0" syn_phrase="PP">
Endlose
</t>
<t g2p_method="lexicon" pos="PTKVZ" sampa="hI-'naUs"
    syn_attach="1" syn_phrase="_">
hinaus
</t>
<t pos="$," syn_attach="2" syn_phrase="_">
,
</t>
<boundary breakindex="4" tone="h-l%"/>
</phrase>
⋮
</div>
⋮
</phonology>
</speaker>
</maryxml>
```

Table 4.3: **Example XML code.** XML representation of the first two lines of the poem "*Die zwei Parallelen*" (cf. Table 4.2 and Section 4.3.6).

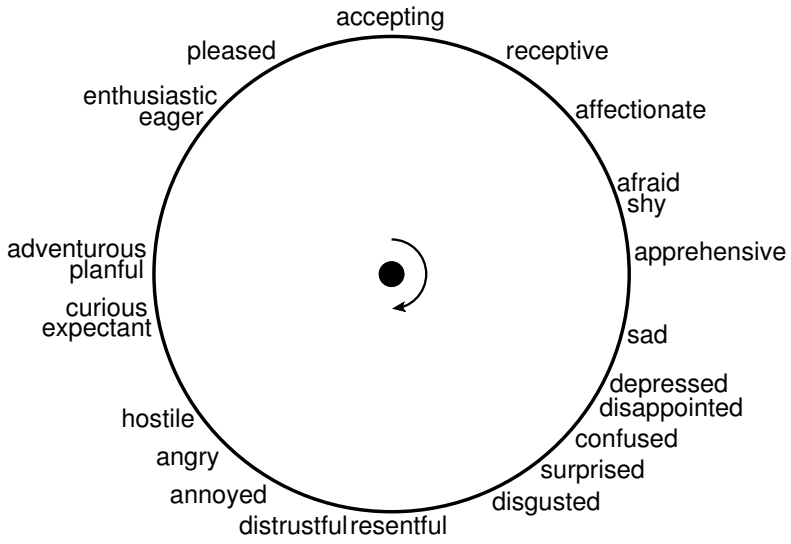


Figure 4.8: **Emotion wheel.** Plutchik [Plu80] arranged emotions according to relative similarity. He found the arrangement to form a circle. Angles between pairs of emotions are a measure for similarity. Arbitrarily, acceptance was chosen to be at 0° of the circle.

used for the description of affective states [CC03, Sch00, HUM05], much work on the expression of emotion has been limited to simple representations such as basic emotions (see e.g. [Sch01]). Only recently, more flexible emotion representations have started to be explored in the domain of speech synthesis [Sch04a] and MPEG-4-based facial animation [TRK+02].
The work presented in this section [ASHS05] follows this line of development in proposing a model for the integrated generation of speech and facial expression using the expressive text-to-speech system MARY in combination with a physics-based facial animation model within MEDUSA. A representation of emotional states combining categorical and dimensional aspects is used for the prediction of vocal and facial expression of non-basic emotions, i.e. of low-intensity and intermediate emotional states.

4.4.1 Emotion Representations

Modeling of emotional expression needs to start from a suitable representation of the emotional states to be expressed.

Emotion categories

The most straightforward description of emotions is the use of emotion-denoting words, or category labels. Human languages have proven to be extremely powerful in producing labels for emotional states: lists of emotion-denoting adjectives were compiled that include at least 107 items [Whi89]. Several approaches exist for reducing these to an essential core set, the most used in the literature being basic emotions, a Darwinian concept [CC03]. Based on the work by Ekman [EK97], basic emotions are usually used for modeling facial expressions of emotions.

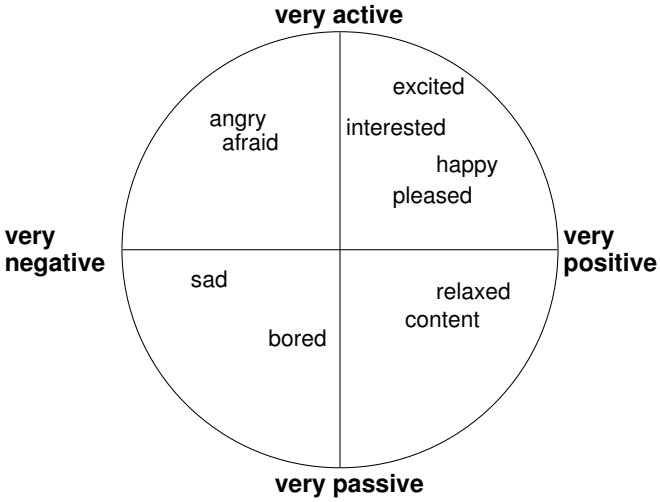


Figure 4.9: **Emotion disk.** Cowie et al. [CDCS$^+$00] proposed a two-dimensional, disk-shaped emotion space with activation and evaluation constituting the axes.

Emotion dimensions

Many different approaches reported in the psychological literature have led to the proposal of dimensions underlying emotional concepts (see [Sch04b] for an overview). Different researchers came to propose two essential dimensions: *activation* (from active / aroused to passive / relaxed) and *evaluation* (from negative / bad to positive / good), sometimes complemented by a third dimension, *power* (from powerful / dominant to weak / submissive). These emotion dimensions are gradual in nature and represent the essential aspects of emotion concepts rather than the fine specifications of individual emotion categories. The names used for these dimensions were selected by the individual researchers *interpreting* their data, and did not arise from the data itself. This explains the large variation found in the literature regarding the names of the dimensions.
One concrete proposal for an emotion dimension model is the activation-evaluation space, proposed by Cowie et al. [CDCS$^+$00]. In accordance to Plutchik's emotion wheel [Plu80] (Figure 4.8), they conceived of the space as circular; but they complemented the circle by a disk whose outer bounds represent maximally intense emotions, while its center (the origin of the two-dimensional space) represents a "neutral", unemotional state. The further a state is from the center, the more intense it is, i.e. the radial distance from the center is a measure of emotion intensity (see Figure 4.9). In accordance to Whissell [Whi89] (Figure 4.10), emotion categories can be located in that space.

Requirements for a natural emotionally expressive system

Databases of naturally occurring emotions [DCCCR03] show that humans usually express low-intensity rather than fullblown emotions, and complex, mixed emotions rather than mere basic emotions downscaled to a low intensity. A system intended to simulate this kind of expressivity needs to use an emotion representation capable of representing such states. Emotion dimensions are a suitable representation: they are naturally gradual, and are capable of representing low-intensity as well as high-intensity states. While they do not define the exact properties of an emotional state in as much detail as a category label, they do capture its essential aspects.

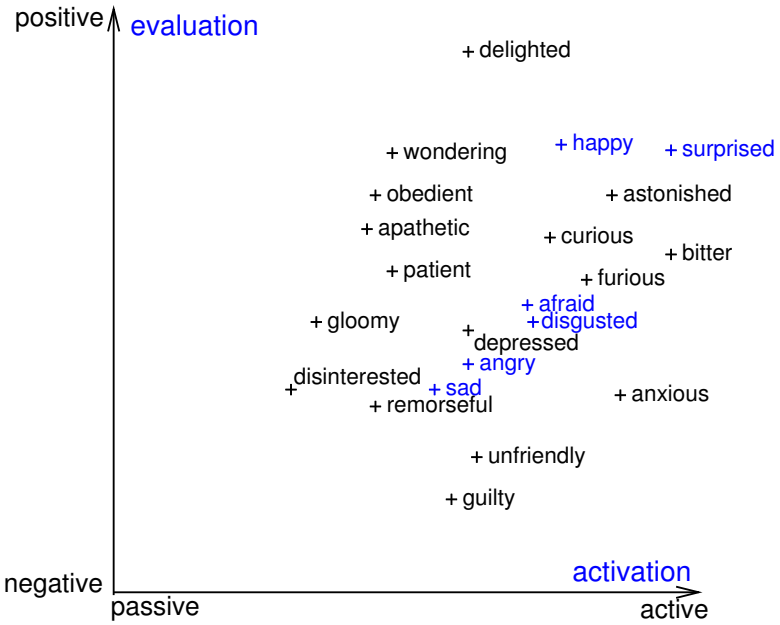


Figure 4.10: **Emotion space.** According to Whissell [Whi89], emotions can be parameterized by their activation and evaluation scores. Blue emotions are Ekman's six basic emotions (e.g. [EK97]).

Mappings between emotion representations

Emotion categories can be *located* in emotion dimension space via rating tests [CDCA+99]. The mapping from categories to dimensions is therefore a simple task, as long as the coordinates of the emotion category have been determined. The inverse, however, is not possible: as emotion dimensions only capture the most essential aspects of an emotion concept, they provide an underspecified description of an emotional state. For example, the coordinates for anger and disgust are very close, because the two categories share the same activation / evaluation / power properties. The features distinguishing between the two categories cannot be represented using emotion dimensions, so that the corresponding region in space can only be mapped to "anger-or-disgust" rather than a specific category. One concrete proposal for a mapping from a list of emotion categories to emotion dimensions was brought forward as a working model by the Net Environment for Embodied Emotional Conversational Agents (NECA) project [KPG+02, NEC05] (see Table 4.4). The NECA project investigates the concept of multi-modal communication with and between virtual characters. Special emphasis is put on personality traits and affective behavior. To this end, research from a variety of different fields, such as natural language generation, speech synthesis, semiotics of non-verbal expression, and emotion as well as personality modeling, is integrated.
It should be kept in mind that, given methodological issues [CC03] as well as the limited empirical basis in existing studies [Whi89, CDCA+99, DCCCR03], mappings between the currently existing emotion representations are necessarily imperfect.

category	activation	evaluation	power
joy	17.3	42.2	12.5
distress	-17.2	-40.1	-52.4
happy-for	17.3	42.2	12.5
gloating	40.0	30.0	30.0
resentment	0.0	-40.0	-20.0
sorry-for	-17.2	-40.1	-52.4
hope	20.0	20.0	-10.0
fear	14.8	-44.4	-79.4
satisfaction	-14.9	33.1	12.2
relief	3.0	33.0	-3.0
fears-confirmed	-30.0	-50.0	-70.0
disappointment	2.4	-24.9	-37.2
pride	30.0	40.0	30.0
admiration	27.0	53.0	17.0
shame	4.6	-26.3	-62.3
reproach	-3.0	-30.0	43.0
liking	-14.9	33.1	12.2
disliking	15.0	-35.0	-10.0
gratitude	20.0	40.0	-30.0
anger	34.0	-35.6	20.0
gratification	-14.9	33.1	12.2
remorse	4.6	-26.3	-62.3
love	1.2	33.3	14.9
hate	60.0	-60.0	30.0

Table 4.4: **Categories in emotion space.** Coordinates for a list of emotion categories on the three emotion dimensions activation, evaluation and power, as proposed as a first working model by the NECA project. All scales range from -100 (passive / negative / submissive) via 0 (neutral) to +100 (active / positive / dominant).

4.4.2 The Emotional Component of the MARY Text-to-Speech System

The MARY TTS system [ST03] has already been introduced in Section 4.3.1. In its more recent form, it is also capable of producing emotional audible speech [Sch04b].
All emotional prosody rules are integrated in a collective module. It adds appropriate annotations to the MaryXML text, which are then executed by the other MARY components as described in Section 4.3.1. As a consequence, all of the parameters are global in the sense that they will be applied to all enclosed text. The approach is highly transparent, as the link between emotions and their acoustic realizations is not hidden in various processing components, and it is easy to maintain and adapt, as all rules are contained in one document.
Since the module is based on linking emotion dimensions to their acoustic correlates, it integrates well with our approach to visual expression modeling presented in Section 4.4.3.
The key properties of MARY's emotional component are reported below.

Emotional prosody rules

Schröder [Sch04b] formulated emotional prosody rules on the basis of a literature review and a database analysis. His literature review brought about the following results. An unambiguous agreement exists concerning the link between the activation dimension and the most frequently measured acoustic parameters: activation is positively correlated with mean **F0**, mean intensity, and, in most cases, with speech rate. Additional parameters positively correlated with activation are pitch range, "blaring" timbre, high-frequency energy, late intensity peaks, intensity increase during a "sense unit", and the slope of **F0** rises between syllable maxima. Higher activation also corresponds to shorter pauses and shorter inter-pause and inter-breath stretches.
The evidence for evaluation and power is less stable. There seems to be a tendency that studies which take only a small number of acoustic parameters into account do not find any acoustic correlates of evaluation and / or power.
The limited evidence regarding the vocal correlates of power indicates that power is basically recognized from the same parameter settings as activation (high tempo, high **F0**, more high-frequency energy, short or few pauses, large intensity range, steep **F0** slope), except that sometimes, high power is correlated with lower **F0** instead of higher **F0**, and power is correlated with vowel duration.
There is even less evidence regarding the acoustic correlates of evaluation. Positive evaluation seems to correspond to a faster speaking rate, less high-frequency energy, low pitch and large pitch range, a "warm" voice quality, longer vowel durations, and the absence of intensity increase within a "sense unit".
In a statistical analysis of the Belfast Naturalistic Emotion Database [DCCCR03], perceptual ratings of the emotion dimensions activation, evaluation and power were correlated with acoustic measures (see [Sch04b, Sch01] for details). The study replicated the basic patterns of correlations between emotion dimensions and acoustic variables. It was shown that the acoustic correlates of the activation dimension were highly stable, while correlates of evaluation and power were smaller in number and magnitude and showed a high variability between male and female speakers. In addition, the analysis provided numerical linear regression coefficients which were used as a starting point for the formulation of quantified emotion prosody rules.
The effects found in the literature and in the database analysis were formulated in a quantified way (Table 4.5) and implemented in the MARY TTS system.
In Table 4.5, the columns represent the emotion dimensions, while the rows list all the acoustic parameters for which emotion effects are modeled. The numeric data fields represent the linear

	Prosodic parameter	Coefficients		
		Activation	**Evaluation**	**Power**
fundamental frequency	pitch	0.3	0.1	-0.1
	pitch-dynamics	0.3%		-0.3%
	range	0.4		
	range-dynamics	1.2%		0.4%
	accent-prominence	0.5%	-0.5%	
	preferred-accent-shape		E≤-20: falling -20<E≤40: rising E>40: alternating	
	accent-slope	1%	-0.5%	
	preferred-boundary-type			P≤0: high P>0: low
tempo	rate	0.5%	0.2%	
	number-of-pauses	0.7%		
	pause-duration	-0.2%		
	vowel-duration		0.3%	0.3%
	nasal-duration		0.3%	0.3%
	liquid-duration		0.3%	0.3%
	plosive-duration	0.5%	-0.3%	
	fricative-duration	0.5%	-0.3%	
	volume	0.33%		

Table 4.5: **Emotion dimension prosody rules.** Values on emotion dimensions range from -100 to 100, with 0 being the "neutral" value. The percentage values are factors – see Section 4.4.2 for details. Source: [Sch04b].

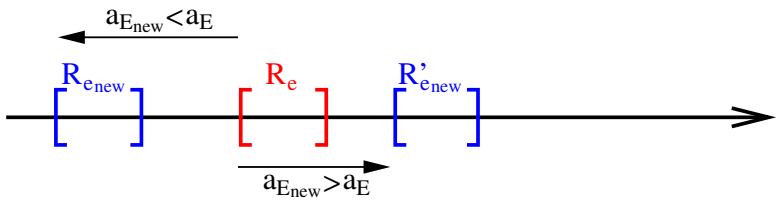


Figure 4.11: **Scaled expressions.** If emotion E and E_{new} differ only in their activation values, then the new expression e_{new} can be computed from e by scaling and shifting its parameter ranges by the ratio of the activation values $\frac{a_{E_{\text{new}}}}{a_E}$ (see Equation (4.1)).

coefficients quantifying the effect of the given emotion dimension on the acoustic parameter, i.e. the change from the neutral default value. As an example, the value 0.5% linking *activation* to *rate* means that for an activation level of +50, rate increases by +25%, while for an activation level of -30, rate decreases by -15%.
The system was evaluated using a perception test. Its results indicate that the speech synthesis system succeeded in expressing the activation dimension (the speaker "arousal"), but not the evaluation dimension. See [Sch04b] for a full account of the experiment.

4.4.3 Intermediate Facial Expressions of Emotion

The human face is capable of displaying many more emotional expressions than just those of the six universal emotions joy, anger, fear, disgust, sadness, and surprise. However, little visual data is available on expressions of other emotions, and modeling them is hard, since differences between them are often subtle. Hence, Tsapatsoulis et al. [TRK+02] have developed a method to interpolate between affect displays to create new ones. We present their original work before we describe our own model derived from theirs.

Tsapatsoulis et al. [TRK+02] modeled emotions using a combination of two emotion models: those by Plutchik [Plu80] and by Whissell [Whi89]. Plutchik ordered 142 emotion words according to their similarity. He found that they can be arranged around a circle, the so-called *emotion wheel* (cf. Figure 4.8). Hence the relative position of each emotion can be described by an angle [Plu80, p.170]. This model does not consider activation or intensity, its goal was to establish similarity. In [Whi89], Whissell describes the second model, a rating of emotion words according to their coordinates on the activation and evaluation dimensions (see Figure 4.10). Tsapatsoulis et al. use the angles in the emotion wheel as a measure of similarity, while they use Whissell's activation values to describe emotion intensity.
Their head model conforms to the MPEG-4 facial animation standard, i.e. it is animated by facial animation parameters (FAPs). Tsapatsoulis et al. identified eight fundamental emotions: acceptance, fear, surprise, sadness, disgust, anger, anticipation, and joy. These are the starting points for the interpolation. The facial expression e corresponding to an emotion E is described by the following parameters:

- the activation value a_E of the emotion E
- its angle on the emotion wheel ω_E
- the set of FAPs F_e involved in forming the expression
- for each contributing FAP $f \in F_e$ the range of variations of its value $R_e(f)$ associated with the expression.

There are two different ways to generate new expressions: if the new emotion E_{new} is very similar to the fundamental emotion E, i.e. if their facial expressions differ mainly in strength of muscle contraction, then the new expression e_{new} can be computed from the expression e in the following way (see also Figure 4.11):

$$\begin{aligned} F_{e_{\text{new}}} &= F_e \\ R_{e_{\text{new}}}(f) &= \frac{a_{E_{\text{new}}}}{a_E} \cdot R_e(f) \quad \forall\, f \in F_e\ . \end{aligned}$$

If the new emotion E_{new} does not clearly belong to a fundamental category, its facial expression is computed by interpolation between the shifted expressions of the two emotions E_1 and E_2 that are neighbors to E_{new} on the emotion wheel. For an interval $I = [i_1, i_2]$, let

$$\sigma(I) = \begin{cases} 1, & i_1 \le i_2 \\ -1, & i_1 > i_2 \end{cases}$$

define the sign σ of I. Let $c(I)$ be the center of interval I and $s(I)$ be its length. Then e_{new} is determined by

$$\begin{aligned} F_{e_{\text{new}}} &= F_{e_1} \cup F_{e_2} \qquad (4.1) \\ R'_{e_1}(f) &= \frac{a_{E_{\text{new}}}}{a_{E_1}} \cdot R_{e_1}(f) \\ R'_{e_2}(f) &= \frac{a_{E_{\text{new}}}}{a_{E_2}} \cdot R_{e_2}(f) \\ c(R_{e_{\text{new}}}(f)) &= \frac{\omega_{E_{\text{new}}} - \omega_{E_1}}{\omega_{E_2} - \omega_{E_1}} \cdot c(R'_{e_2}(f)) \ + \ \frac{\omega_{E_2} - \omega_{E_{\text{new}}}}{\omega_{E_2} - \omega_{E_1}} \cdot c(R'_{e_1}(f)) \\ s(R_{e_{\text{new}}}(f)) &= \frac{\omega_{E_{\text{new}}} - \omega_{E_1}}{\omega_{E_2} - \omega_{E_1}} \cdot s(R'_{e_2}(f)) \ + \ \frac{\omega_{E_2} - \omega_{E_{\text{new}}}}{\omega_{E_2} - \omega_{E_1}} \cdot s(R'_{e_1}(f)) \end{aligned}$$

$$R_{e_{\text{new}}}(f) = \begin{cases} \left[c(R_{e_{\text{new}}}(f)) - \frac{1}{2} \cdot s(R_{e_{\text{new}}}(f)),\ c(R_{e_{\text{new}}}(f)) + \frac{1}{2} \cdot s(R_{e_{\text{new}}}(f))\right] & \\ \qquad \forall\, f \in F_{e_1} \cap F_{e_2} : \sigma(R_{e_1}(f)) = \sigma(R_{e_2}(f)) & (4.2) \\ \frac{a_{E_{\text{new}}}}{a_{E_1}} \cdot R_{e_1}(f) \ \cap \ \frac{a_{E_{\text{new}}}}{a_{E_2}} \cdot R_{e_2}(f) & \\ \qquad \forall\, f \in F_{e_1} \cap F_{e_2} : \sigma(R_{e_1}(f)) \neq \sigma(R_{e_2}(f)) & (4.3) \\ \frac{a_{E_{\text{new}}}}{2 \cdot a_{E_1}} \cdot R_{e_1}(f) \quad \forall\, f \in F_{e_1} \setminus F_{e_2} & (4.4) \\ \frac{a_{E_{\text{new}}}}{2 \cdot a_{E_2}} \cdot R_{e_2}(f) \quad \forall\, f \in F_{e_2} \setminus F_{e_1}\ . & (4.5) \end{cases}$$

Now, for all $f \in F_{e_{\text{new}}}$ with $R_{e_{\text{new}}}(f) = \emptyset$ set $F_{e_{\text{new}}} := F_{e_{\text{new}}} \setminus \{f\}$.
In case FAP f is involved in the facial expressions of both generating emotions and its variation intervals have for both emotions the same sign (Equation (4.2)), i.e. it describes movement into the same direction, the variation intervals of the generating expressions e_1 and e_2 are first scaled and shifted, so that the resulting expressions have the same activation as E_{new} (Figure 4.12 top).

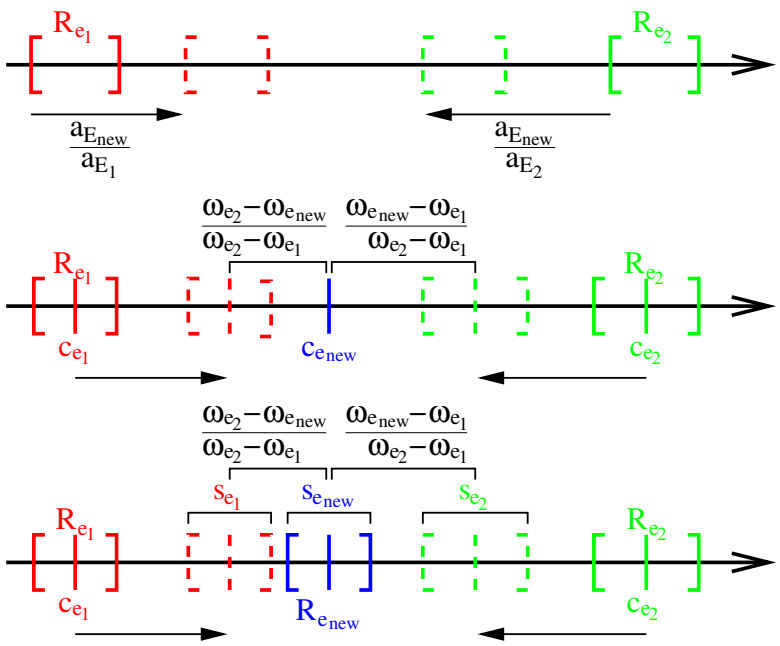


Figure 4.12: **Intermediate expressions without parameter conflicts.** Top: the range of each FAP in the generating expressions e_1 and e_2 is first scaled and shifted according to the ratios of the activation values $\frac{a_{E_{\text{new}}}}{a_{E_1}}$ and $\frac{a_{E_{\text{new}}}}{a_{E_2}}$. Then, the shifted ranges are linearly interpolated, parameterized by the emotion angles $\omega_{E_{\text{new}}}$, ω_{E_1} and ω_{E_2}. Center: the center $c_{e_{\text{new}}} := c(R_{e_{\text{new}}})$ of the new parameter range is determined from $c_{e_1} := c(R_{e_1})$ and $c_{e_2} := c(R_{e_2})$. Bottom: in the same way, the length $s_{e_{\text{new}}} := s(R_{e_{\text{new}}})$ of interval $R_{e_{\text{new}}}$ is computed to yield the new expression e_{new}. See also Equation (4.2).

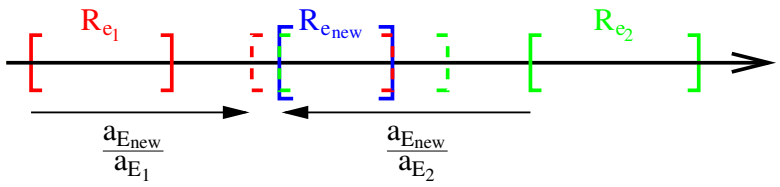


Figure 4.13: **Intermediate expressions with parameter conflicts.** In case a parameter is present in both generating facial expressions e_1 and e_2, but acts in opposite directions, Equation (4.3) applies. The ranges R_{e_1} and R_{e_2} of each FAP are scaled and shifted according to the ratios of the activation values $\frac{a_{E_{\text{new}}}}{a_{E_1}}$ and $\frac{a_{E_{\text{new}}}}{a_{E_2}}$. The shifted ranges are then intersected to yield the range $R_{e_{\text{new}}}$ for this FAP for the new expression e_{new}.

Then from the centers and lengths of the shifted intervals the interval $R_{e_{\text{new}}}(f)$ can be computed through linear interpolation between the emotion wheel angles as depicted in Figure 4.12 (center and bottom).
If the variation intervals of e_1 and e_2 have different signs (Equation (4.3)), the interval of the new expression is the intersection of the original ones, see Figure 4.13.
If f is present only in one generating expression (Equations (4.4) and (4.5)), say, e_1, then its variation interval is averaged with the interval of the neutral face e_0, for which $a_{E_0} = 0$ and $R_{e_0}(f) = [0]$ for all FAPs f.

zygomaticus major left	$\longleftrightarrow$	*depressor anguli oris left*
zygomaticus major right	$\longleftrightarrow$	*depressor anguli oris right*
orbicularis oris	$\longleftrightarrow$	{*risorius left*, *risorius right*}
mentalis	$\longleftrightarrow$	{*depressor labii inferioris left*, *depressor labii inferioris right*}

Table 4.6: **Antagonists.** Facial muscles operating in a roughly antagonistic fashion.

We have modified the approach to work with our physics-based model. Instead of combining the data from the Whissell and Plutchik studies, as Tsapatsoulis et al. did, we use the set of emotion words with associated coordinates on the three dimensions activation, evaluation and power, as proposed by the NECA project [KPG+02] (see Table 4.4). In this first version of the system, we only use the first two dimensions from this table (see Section 4.4.4).
We use Cowie et al.'s disk-shaped activation-evaluation space (see Figure 4.9) as our model of emotion dimensions. It appears natural to describe the states in the activation-evaluation space by means of polar coordinates, using angular orientation ω and radial distance from the center r. Here again, the angle ω describes similarity. In contrast to Tsapatsoulis et al., we consider radial distance from the center of the activation-evaluation space to be a better indicator of emotional intensity than activation (consider the case of despair, which would have high intensity but low activation), and therefore use this radial distance r rather than the activation level a for normalising the archetypal states' intensities to the intermediate state's intensity in our equations.
As our "basic" emotions, we use the closest correlates to the six Ekmanian emotions (joy, anger, fear, disgust, sadness, and surprise) that we can find in Table 4.4: joy, anger, fear, hate, sorry-for, and surprise (as a state with 100% activation, and 0% on evaluation and power). We are aware that these are crude approximations, which should be taken as illustrating the idea rather than as a final truth.
Since our animations are based mostly on muscle contractions instead of MPEG-4 FAPs, we had to adapt the approach to also work with muscles. We defined our expressions through single muscle contraction values $v \in [0, 1]$. They can be uniformly scaled by a number between 0 and 1 to achieve different intensities of the expressed emotion, but we leave it to the animator to decide how small the scaling value can be so that the resulting expression is still perceived as the same emotion. As a consequence, we have no means of deciding for a given muscle whether to use Equation (4.2) or Equation (4.3). Hence, we identified facial muscles that operate in a roughly antagonistic fashion (see Table 4.6).
Let M_e be the set of muscles involved in expression e of emotion E and $v_e(m)$ the contraction value of $m \in M_e$. For simplicity, the animation parameters of eyes, tongue, jaw, and head are included in M_e. This leads to the following modified algorithm for the case where the new emotion E_{new} is similar to a fundamental emotion E (see Figure 4.14):

$$M_{e_{\text{new}}} = M_e \tag{4.6}$$

$$v_{e_{\text{new}}}(m) = \frac{r_{E_{\text{new}}}}{r_E} \cdot v_e(m) \quad \forall\, m \in M_e\ . \tag{4.7}$$

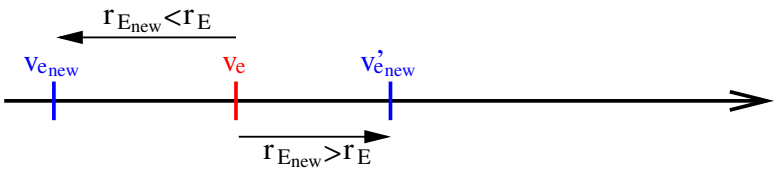


Figure 4.14: **Scaled expressions.** If the new expression e_{new} is a weaker or stronger version of an already existing expression e, the contraction value v of each muscle is scaled by the ratio of the activations $\frac{a_{E_{\text{new}}}}{a_E}$, see Equation (4.7).

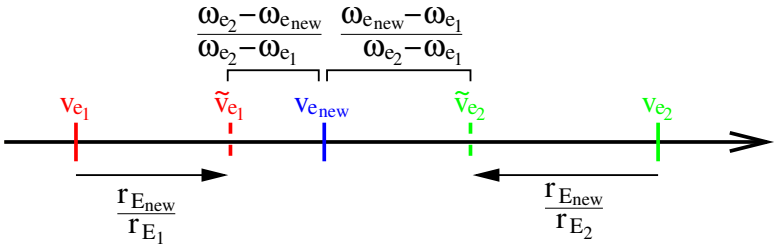


Figure 4.15: **Muscle contraction for intermediate expressions for a muscle without antagonist.** In accordance with Equation (4.9), the muscle contraction values for muscle m in the generatign epxressions e_1 and e_2, i.e. $v_{e_1}(m)$ and $v_{e_2}(m)$, are scaled by the ratio of the emotion disc radii $\frac{r_{E_{\text{new}}}}{r_{E_1}}$ and $\frac{r_{E_{\text{new}}}}{r_{E_2}}$. Linear interpolation between the new values $\tilde{v}_{e_1}(m)$ and $\tilde{v}_{e_2}(m)$, parameterized by the angles on the emotion disc, yields the contraction value $v_{e_{\text{new}}}(m)$ for muscle m in the new expression e_{new}.

Since several muscles can be antagonistic to others, e.g. the *orbicularis oris* to both the *risorius left* and the *risorius right*, we define for every muscle m the set of its antagonists as $A_-(m)$ and the set of muscles that share these antagonists as $A_+(m)$. For m = *risorus left* for instance, $A_+(m) = \{$*risorius left, risorius right*$\}$ and $A_-(m) = \{$*orbicularis oris*$\}$. If the facial expression for E_{new} is computed from two fundamental expressions E_1 and E_2, we get:

$$
\begin{aligned}
M_{e_{\text{new}}} &= M_{e_1} \cup M_{e_2} && (4.8)\\
v'_{e_1}(m) &= \frac{r_{E_{\text{new}}}}{r_{E_1}} \cdot v_{e_1}(m)\\
v'_{e_2}(m) &= \frac{r_{E_{\text{new}}}}{r_{E_2}} \cdot v_{e_2}(m)\\
v_{e_{\text{new}}}(m) &= \frac{\omega_{E_{\text{new}}} - \omega_{E_1}}{\omega_{E_2} - \omega_{E_1}} \cdot v'_{e_2}(m) \;+\; \frac{\omega_{E_2} - \omega_{E_{\text{new}}}}{\omega_{E_2} - \omega_{E_1}} \cdot v'_{e_1}(m)\\
&\quad \forall\, m \in M_{e_{\text{new}}} :\; A_-(m) = \emptyset && (4.9)\\
S_+ &= \sum_{m' \in A_+(m)} \left(\frac{r_{E_{\text{new}}}}{r_{E_1}} \cdot v_{e_1}(m') \;+\; \frac{r_{E_{\text{new}}}}{r_{E_2}} \cdot v_{e_2}(m') \right)\\
S_- &= \sum_{m' \in A_-(m)} \left(\frac{r_{E_{\text{new}}}}{r_{E_1}} \cdot v_{e_1}(m') \;+\; \frac{r_{E_{\text{new}}}}{r_{E_2}} \cdot v_{e_2}(m') \right)
\end{aligned}
$$

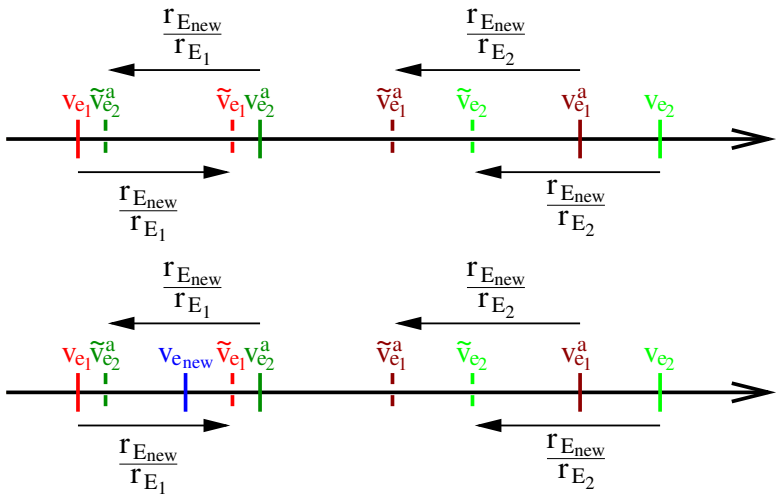


Figure 4.16: **Muscle contraction for intermediate expressions for a muscle with antagonist.** Top: in this case, not only the contraction values of muscle m for both generating expressions e_1 and e_2 are scaled by the ratio of the radii of on the emotion disc, but also the contraction values $v^a_{e_1} := v_{e_1}(m^a)$ and $v^a_{e_2} := v_{e_2}(m^a)$ of m's antagonistic muscle m^a (Equation (4.10)). Bottom: if the sum of the shifted contractions of the muscle is greater than that of its antagonist, i.e. $\tilde{v}_{e_1}(m) + \tilde{v}_{e_2}(m) > \tilde{v}_{e_1}(m^a) + \tilde{v}_{e_2}(m^a)$, then the new contraction value $v_{e_{\text{new}}}$ for m is computed as the difference between the shifted contraction values of the muscle and the shifted contraction values of the antagonist: $v_{e_{\text{new}}}(m) = (\tilde{v}_{e_1}(m) + \tilde{v}_{e_2}(m)) - (\tilde{v}_{e_1}(m^a) + \tilde{v}_{e_2}(m^a))$.

$$v_{e_{\text{new}}}(m) = \begin{cases} 0, & \text{if } S_+ \leq S_- \\ (S_+ - S_-) \cdot \frac{1}{S_+} \cdot \left(\frac{r_{E_{\text{new}}}}{r_{E_1}} \cdot v_{e_1}(m) + \frac{r_{E_{\text{new}}}}{r_{E_2}} \cdot v_{e_2}(m) \right), & \text{else} \end{cases}$$
$$\forall\, m \in M_{e_{\text{new}}} : \; A_-(m) \neq \emptyset \tag{4.10}$$

Equation (4.8) is analogue to Equation (4.1). The differences in Equation (4.9) result from the use of a single value instead of an interval. This obviates the need to compute the center and length of the interval. Instead we can scale and interpolate the contraction values directly (Figure 4.15). Since they do not describe a direction but a value, no conflict arises. The main difference lies in Equation (4.10). $S_+(m)$ and $S_-(m)$ are the summed, scaled contraction values of all muscles in the same and "opposite" antagonistic class, respectively. The overall scaled contraction for all muscles in the same and the antagonistic class of m is $S_+(m) - S_-(m)$. This is distributed to the individual muscles of the set with the stronger scaled overall contraction according to their contribution to that value (see Figure 4.16). If we assign contraction values $v_{e_1}(m) = 0 \quad \forall m \in M_{e_2} \setminus M_{e_1}$ and $v_{e_2}(m) = 0 \quad \forall m \in M_{e_1} \setminus M_{e_2}$, this obviates the need for Equations (4.4) and (4.5).

In Figure 4.17, the new expressions *anxiety* and *panic fear* have been generated as a scaled version of *fear*, following the method in Equations (4.6) and (4.7). In the examples in Figures 4.18 and 4.19, new expressions (center of each row), have been generated from the fundamental expressions to the left and right as described in Equations (4.8) to (4.10).

anxiety	**fear**	**panic fear**
a = 8	a = 14.8	a = 20
e = -24.1	e = -44.4	e = -60.1
r = 25.3	r = 46.8	r = 63.4
ω = 288.4	ω = 288.4	ω = 288.4

Figure 4.17: **Scaling emotions.** *Anxiety* and *panic* belong to the same fundamental class as *fear*, but differ in intensity. Therefore their facial expressions can be generated from fear by scaling by the ratio of the radii. The angle on the emotion disc is kept fixed for both new expressions, while the radii are varied, thereby yielding new values for activation and evaluation.

4.4.4 Conclusions

We have presented a flexible approach to generating non-basic, mixed emotional states in the facial expressions of an anatomically based talking head. This has been achieved by modifying the work of Tsapatsoulis et al. [TRK+02], aimed at an MPEG-4-based face model, to a physics-based facial animation system. In extension to their work, we have used data in a single emotion model, the activation-evaluation space [CDCS+00], for indicating both emotion quality and intensity. As a result, our system is able to generate emotional facial expressions of various intensities, and to show mixed emotions by a gradual blending of facial configurations of basic emotions.
We have combined the MEDUSA facial animation system with an emotional text-to-speech synthesis system which is also based on emotion dimensions. In combining these two components, we are able to create photo-realistic animations of a talking head capable of expressing a continuum of shades of emotion.

There are several possible directions for future work. The most exciting is the extension to 3D emotion space, i.e. to not only consider activation and evaluation, but also power to allow for a more fine-grained model. Since emotions are arranged inside a sphere in this space, we propose to project the individual emotions onto the sphere's surface and to interpolate between expressions on the surface of the sphere. This would permit interpolation between more than two expressions. The resulting expression is then projected back to the desired distance from the origin.
A next step should be the evaluation of the system. This could be done in a similar manner as described in [Sch04b].
Since emotion categories are more intuitive for most people than positions in activation-

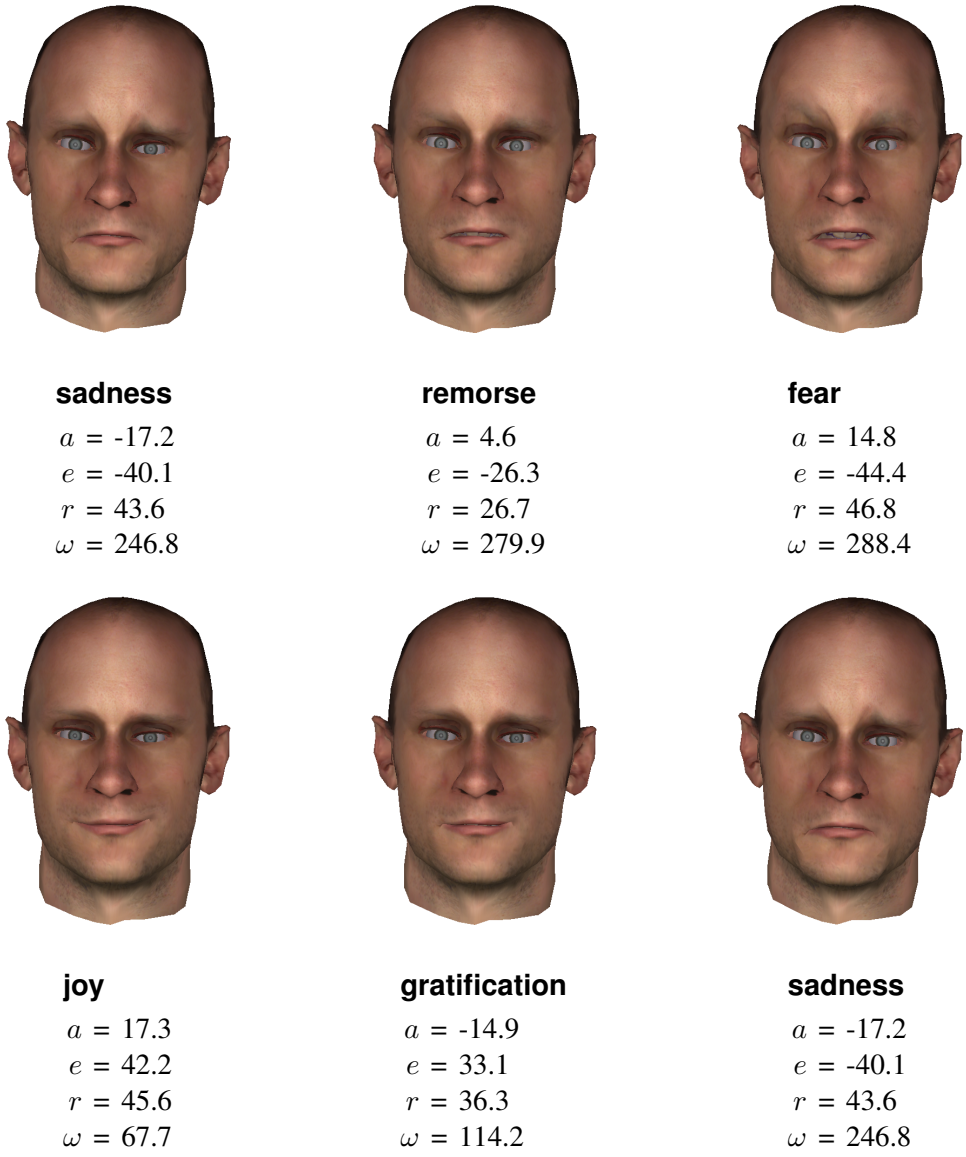


Figure 4.18: **Blending emotions.** The emotional expression in the middle has been obtained from those at the left and right using the blending algorithm. The radius r and the angle on the emotion disc ω determine the influence of each generating expression and hence the degree of similarity to the new one. The coordinates in emotion space have been obtained from the NECA data.

evaluation-power space, we require coordinates for more emotion words to enhance the user-friendliness of the system. Another pressing issue is the extension of the system to include different emotions in a single utterance, allowing for transitions between emotions over time. Adapting the frequency and strength of the non-verbal speech-related facial expressions to the current emotion could enhance the realism of the animations, e.g. look downwards more often and in general show movement with less amplitude when sad. As an additional visible effect of emotion the artificial face should be capable of blushing. Frequency and intensity of breathing are also indicators of the emotion currently felt.

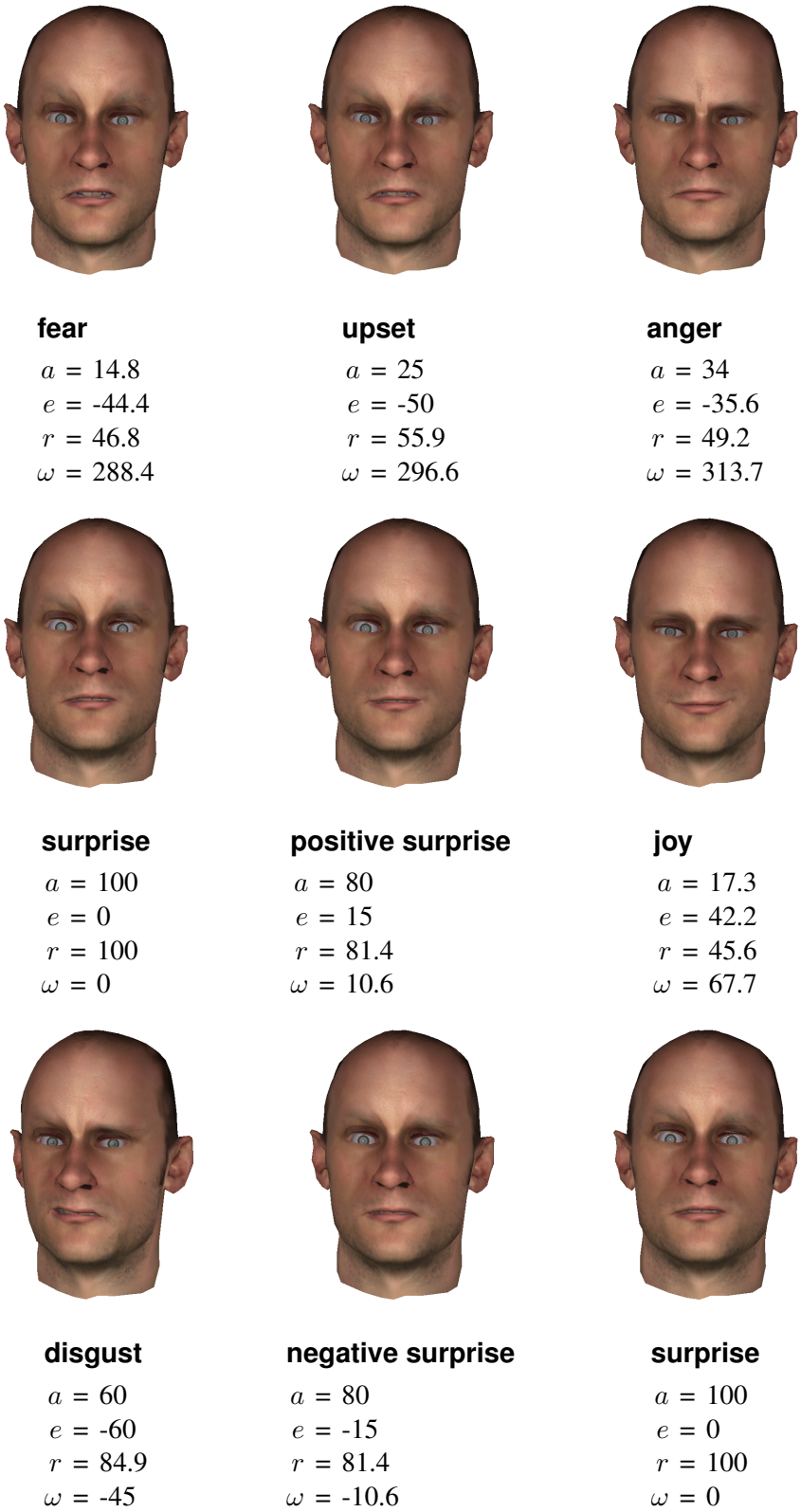


Figure 4.19: **Blending emotions (continued).** See also Figure 4.18. The first example is a not too active, but rather negative emotion, while the second one could be pleasant surprise, and the last one unpleasant surprise.

5

A Facial Composite System

Every once a while, facial composites appear in newspapers or the news. Law enforcement agencies use them to track down eyewitnesses, to gather information about suspects, to ask for help with the identification of unknown bodies, in short, whenever a photograph of the person is not available.
Formerly, a forensic artist sketched the target face according to the witness' description. Sometimes this method is still used, since it is the most flexible one.
In the late 1950's Smith&Wesson® published the Identi-Kit system (Figure 5.1). For every facial feature it contains a set of transparencies with drawings of the respective face part. The witness selects the appropriate slides, from which the face is pieced together in a frame. Every transparency has a unique tag, and frame positions are rasterized. This information could be cabled to other police stations, where an identical composite was assembled. Practice, however, showed that this method had its flaws. Due to the lack of shading and the limited number of slides, resemblance was difficult to obtain. As a result, the forensic artist would quite often draw onto the slides to enhance the portrayal, thereby proving the original idea worthless.
PhotoFit, a similar system where faces were assembled from photographed features, was developed in the 1970's. Later, this methodology was ported to computers and implemented in a variety of facial composite programs (cf. Section 2.2).
Although facial composites have been around for so long, they do not have the significance one

Figure 5.1: **Identi-Kit slide system.** The witness selects slides with individual features which are assembled in a rasterized frame.

might expect. The reason are low recognition rates, at least partially due to the fact that actively recalling faces is extremely difficult. The human brain is better equipped for recognition of faces than for describing them. This deficiency is further aggravated by the stress witnesses experience and the fact that they may only have caught a brief glimpse of the (possibly masked) target person. The result will be that the eyewitness' mental image of the person is vague and cloudy. The goal of every facial composite system must therefore be to help the witness as much as possible with this difficult task.

We let this maxim guide us when we developed our mind2model facial composite system [BAHS]. It leaves the user in full control, but supports him by automatically incorporating statistics of faces. The system considers anatomical and ethnical correlations between features and exploits them to present the user with the most plausible face after every editing step. Features that the witness does not remember at all are filled in automatically to harmonize with the rest of the face. Editing is mostly done by manipulating intuitive attributes, but importing features from a database is also possible. The user always works with a complete, anatomically correct three-dimensional face that can be viewed from all directions and under any lighting conditions.
Since faces from mind2model are three-dimensional and usability is easy, the system also lends itself to creating characters for movies, advertisement, or computer games, where the player can model virtual characters according to his imagination or have them impersonate his personal friends or enemies.
To sum up, both facial composite creation in law enforcement and design of virtual characters share important properties: the artist / witness (denoted as *source* in the following) possesses a more or less clear mental image of the *target* face to be created, where it obviously is important that this mental image must not be modified or diminished during the process of reconstructing the face. Moreover, the source is usually not able to give a complete description of the target face. Typically, some striking details, for instance bushy eyebrows or a hooked nose, are present in the mental images, while other, more subtle characteristics, such as the distance between eyes and eyebrows, are not.

The mind2model facial composite system is presented in Section 5.1. We conducted a user study to evaluate the system. Proceedings and results can be found in Section 5.2. The chapter concludes with a discussion in Section 5.3.

5.1 The mind2model Facial Composite System

The mind2model system was designed for generating models of human faces from vague mental images or incomplete descriptions. In a way, the system is similar to commercial systems for creating composite or photofit pictures as regularly used in police work. Compared to these programs, however, our approach has several significant advantages:

- our GUI offers intuitive ways to modify features of the target face in arbitrary order (see Figures 5.2 (left) and 5.4)
- unspecified parts of the reconstructed face are automatically completed according to statistical properties
- anatomical / ethnical correlations within a face are taken into account automatically during reconstruction

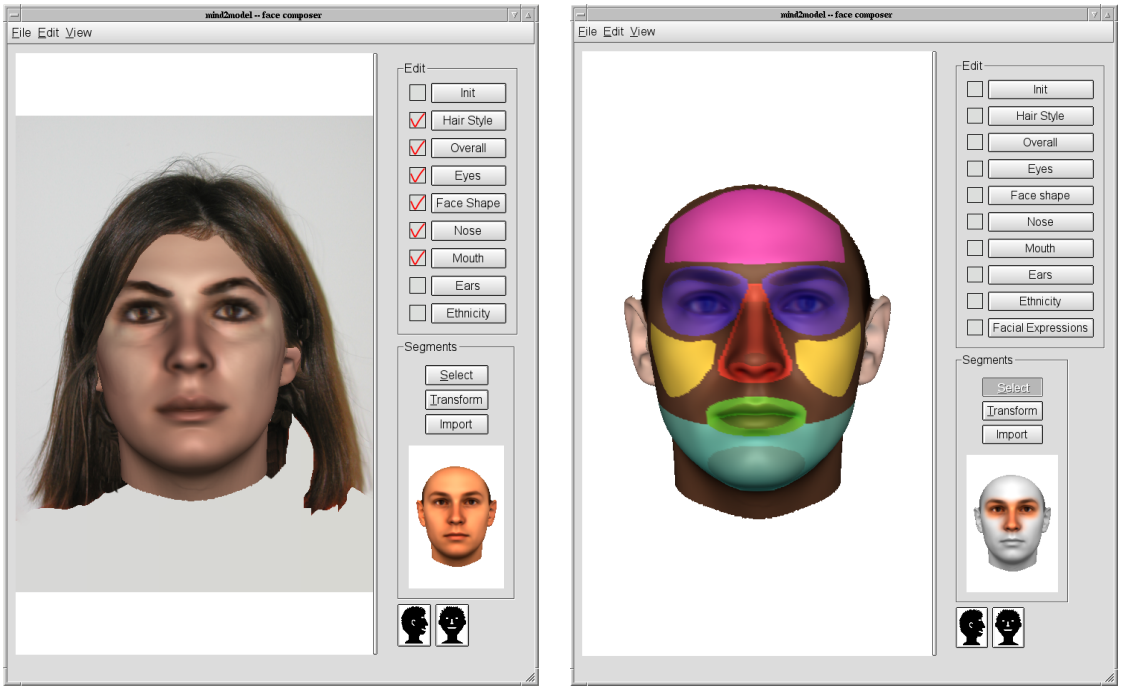
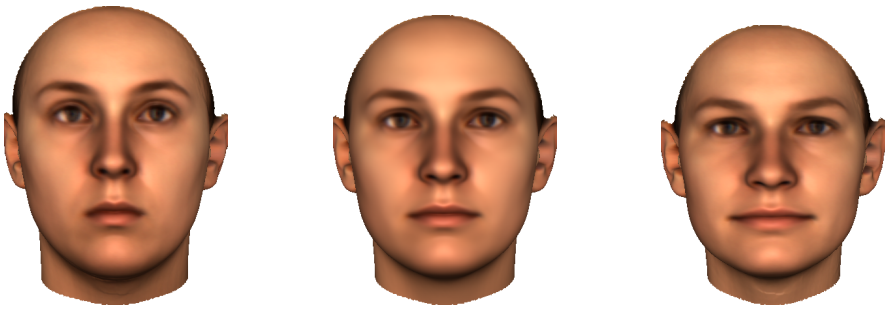


Figure 5.2: **System snapshots.** Left: GUI main window. The composite face is displayed in the large widget on the left. To the right are the buttons leading to the dialogs for attribute editing (cf. Figure 5.4) and the widget group for selecting (see right), transforming, or importing facial features from a database (cf. Figure 5.5). The buttons at the lower right are shortcuts for choosing a frontal or profile view of the composite. Right: in selection mode, a colored mask is laid over the face with individual segments color coded. The user selects segments by clicking on the mask. The current selection is highlighted in the small widget on the right.

- a 3D face model is created that can be viewed from an arbitrary viewpoint under arbitrary lighting conditions
- the resulting face model can be rendered automatically into background images in appropriate pose and illumination.

The technology behind this system is based on a three-dimensional morphable model of faces and extends earlier work by Blanz et al. [BV99, BSVS04] (see also Section 3.2.1). Our algorithm for navigating face space uses a set of attribute constraints that restrict the face to a residual subspace. The prediction of unspecified facial features is based on correlation between different face regions and features learned from the database. Our system makes the most plausible prediction, given the information provided by the user. Photographs of new individuals are used to augment the databases available for the 3D modeling process. Using attribute constraints and face exchange, the database of example faces is automatically adjusted to the user's specifications.

This section gives an overview of the functionality and the features of our system. The editing process starts from the *average* face computed from 100 male and 100 female faces (cf. Section 3.2.1). This initial face is then modified through various editing operations which lead to an immediate update of the displayed face. To facilitate user interaction, we have implemented an operation history with multiple levels of "undo".



narrow mouth female average wide mouth

Figure 5.3: **Correlation between features.** Center: average female face. Left: decreasing the width of the mouth results in a more oval face, a smaller nose, and rounder eyes. Right: making the mouth wider leads to a more angular face, increases the width of nose and eyes, and the eyes appear more lively.

5.1.1 General Settings

As a first step, the user (i.e. either the source or an operator) may specify age, gender, and ethnicity of the target face. This is an optional step, which does not directly affect the face model. Instead, setting these parameters restricts the selection of example faces shown in the dialog box for importing features to match the criteria specified for age, gender, and ethnicity (but see Section 5.1.5).

5.1.2 Segments

Editing operations on one part of the face may also affect other parts due to correlations between individual features and overall face shape. A narrow mouth, for example, correlates with a narrow nose, a more oval face, and rounder eyes, while increasing mouth width will lead to a more angular face, a broader nose, and bigger eyes (Figure 5.3). It is one of the main advantages of our approach that these correlations are taken into account automatically. In cases where only little is known or remembered of the target face, exploiting these correlations will lead to a coherent composite of the most probable face for the given user input, and hence may add significantly to the faithfulness of the reconstruction. Sometimes, however, the source may want to change a single feature only without any effects on the rest of the face. Therefore the effects of editing operations may be constrained to a local area.
For this purpose we divided the face into a shallow hierarchy of *segments* (see Section 3.2.2). All segments are listed in Table 5.1. The root of the hierarchy is the entire face, which is divided into segments corresponding to individual features. Some features are again subdivided into several child segments. Figure 5.2 (right) shows the GUI of our program in selection mode, where different segments are color coded. Child segments are shown in different shades of their parent's color. The nose, for instance, consists of individual segments for the root, the nose bridge, the alar wings, the tip, and the base area of the nose, all marked in different shades of red. In selection mode, the user clicks on one or more segments to add them to the selection mask shown in the small pixmap on the right. The user may add segments from any hierarchy level of any facial feature to the selection. Similarly, segments can be excluded from the mask. This enables the user to restrict edit operations to any combination of segments. To ensure smooth transitions at the segment borders, we apply a frequency-dependent blending technique [BA83].

face							
forehead	eyes	cheeks	ears	nose	mouth	chin	rest
	eyebrows orbits, brow ridge eyes			root bridge wings base tip	upper lip lower lip peripheral oral region	tip jaw	

Table 5.1: **List of segments.** Segments are organized in a three-level hierarchy. The entire face is divided into features, which in turn have several sub-segments. It is possible to select any combination of segments and sub-segments.

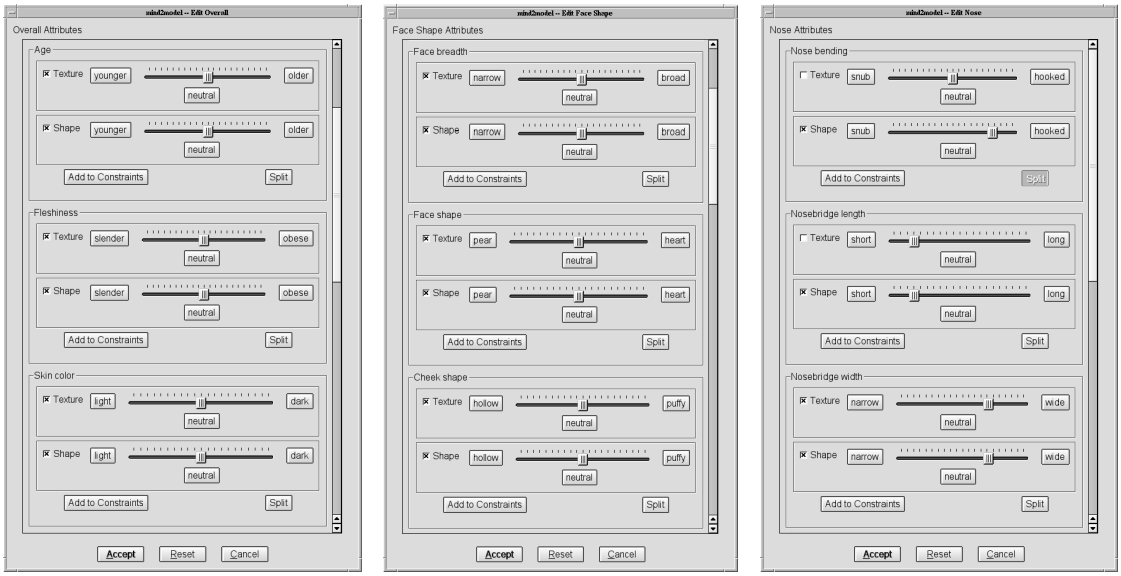


Figure 5.4: **Editing attributes.** By manipulating the sliders, the user modifies particular attributes of the face or selected facial features.

5.1.3 Affine Transformations

Any time during the reconstruction of the target face, affine transformations (rotation, translation, non-uniform scaling) may be applied to the entire face or a currently selected segment. In the case of segments, these transformations are useful, for instance, to modify the proportions and relative positions of features or to account for an asymmetric layout of facial features. Applied to the entire face, they allow the user to adjust the position and orientation of the face to fit into a background image (cf. Section 5.1.8).

5.1.4 Facial Attributes

The majority of editing operations is done by manipulating facial attributes such as face shape, prominence of cheek bones, mouth width, etc. The saliency of each attribute can be set to any value between -1 and 1 on a continuous scale, where 0 is the default value of the average face. We have assigned opposite terms to each side of the scale to describe the effect that moving the

category	attributes
overall	masculine–feminine, slender–obese, dark–light skin, younger–older, intensity of freckles, intensity of beard shadow, attractiveness, caricature level
face shape	round–angular, narrow–broad, pear–heart shape, hollow–puffy cheeks, prominence of cheek bones, pointed–broad chin, receding–protruding chin, distance between lips and chin, intensity of double chin, intensity of nasolabial fold
eyes	slitted–round, upwards–downwards inclination, horizontal distance of eyeballs, dark–light iris, color of iris, dark–light eyebrows, thin–bushy eyebrows, straight–curved eyebrows (separate for left and right side), horizontal distance of eyebrows, distance between eyebrows and eyes
nose	short–long nose bridge, narrow–wide nose bridge, narrow–wide alar wings, flat–round alar wings, snub–hooked nose, distance between nose and mouth
mouth	narrow–wide, thin–full lips, dark–light lip color, convex–concave lip line
ears	small–large, flat–jug ears
ethnicity	Caucasian, Asian, African
expressions	smiling, angry, surprised, scared, deranged, disgusted

Table 5.2: **List of Attributes.** Attributes are divided into categories to facilitate inspection.

attribute value in this direction will have on the face, for example "retracting" and "protruding" chin shapes. The attributes available in the system are listed in Table 5.2. Additional facial attributes can be easily integrated through parameter files. Section 3.2.2 explains how attributes are learned from a database of example faces.

Attributes are pooled in groups according to the facial feature they affect. For example, all facial attributes that affect the shape or color of the nose are grouped together. Figure 5.4 shows GUI dialogs with sliders for attributes of the entire face, attributes describing face shape, and nose attributes. During editing operations, the effect of each facial attribute on the target face (or on the currently selected segment) can be restricted to the shape and / or texture of the target face (selected segment). In addition, different values can be set for the shape and texture parameter of each attribute. Handling shape and texture separately is desirable, for instance, if the user wants to change the skin color only, or to keep the texture homogeneous over the entire face when operations are restricted to a segment.

Of special interest in the context of law enforcement is the *caricature* attribute. It increases the distinctiveness of the face by morphing it away from the average (see Section 3.2.2). Deffenbacher et al. [DCL81] demonstrated that a slightly caricatured version of a face is in general easier to recognize than the original.

As explained in Section 5.1.2, some facial features are correlated to other facial features, i.e. applying an attribute vector to the face may affect previously set values of other attributes.

This is a desired feature of our system, since it makes the sources' life easier, especially when their mental image is incomplete. Sometimes, however, the user wants a specific value for a facial attribute A to remain untouched by further editing operations on other facial attributes. For instance, the distance between eyes and eyebrows is correlated to gender: moving the eyebrows up makes the face appear more feminine. Hence the user might want to protect the masculine appearance of the edit face while increasing the correlated eye–eyebrow distance (see Figure 3.8). This can be achieved by adding the current setting of attribute A to a list of constraints. Section 5.1.7 explains how these constraints are enforced throughout further editing operations.

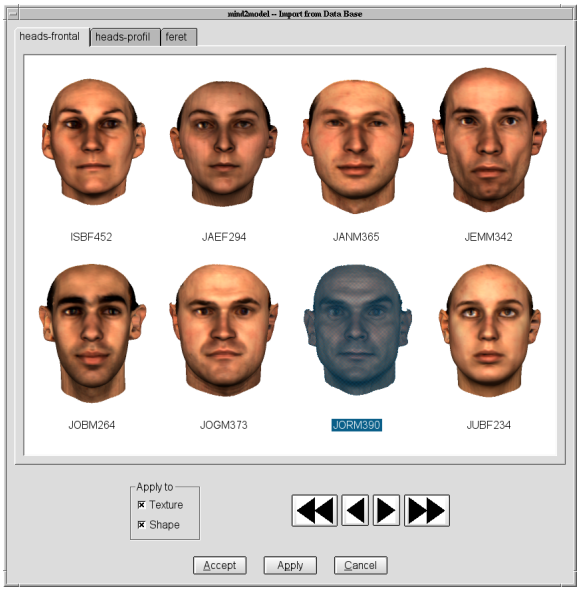


Figure 5.5: **Selecting features.** Either texture, shape, or both is transferred from the selected face in the database to the currently active segments in the edit head.

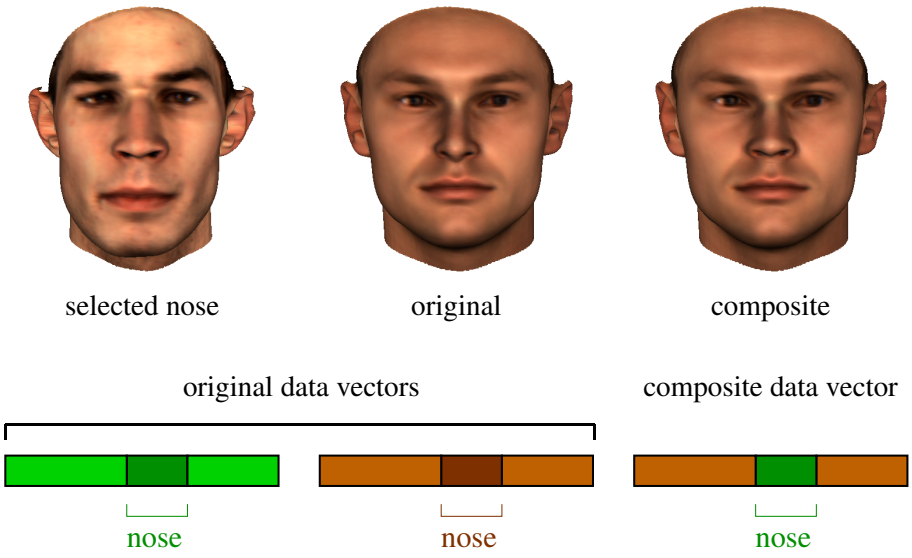


Figure 5.6: **Exchanging features.** The nose of the face in the center was replaced by the nose of the face on the left in order to obtain the composite on the right. The entries corresponding to the nose in the data vector of the left face were substituted for the nose entries of the data vector of the center face. This gives us the combined data vector of the composite.

5.1.5 Importing Features from a Database

Similar to classical composite systems, mind2model also offers the possibility to import features from a database of example faces into the edit face (Figure 5.5). If a face from the database is selected, the parts of the edit face that belong to the current segment mask (Section 5.1.2) are replaced by the corresponding features of the example face. In order to exchange, for instance,

eyes nose all other segments composite

Figure 5.7: **Adaptation to local populations.** Although there was no African face in the model database, fitting the morphable model to faces of African Americans and assembling features from these new faces yields realistic results. For the composite on the right, the eyes were selected from the face in the first photograph, the nose comes from the second photograph, and the remaining segments were taken from the face in the third picture.

the nose of the edit face, the user updates the selection mask to contain only the nose segment and then selects the face with the desired nose from the example database. The nose of that face will be imported seamlessly into the composite (Figure 5.6) and can be further edited if desired. Since all face models are in correspondence, the substitution is achieved by replacing those entries in the shape and texture vectors of the edit face that correspond to the currently active segments by the respective entries of the data vectors of the example face. Hence the data vectors of the new face are combinations of the vectors of the input faces. At the segment boundaries, we perform blending as mentioned in Section 5.1.2. As with attribute editing, the importing of features from a database can be restricted to shape and / or texture.

5.1.6 Adapting the Database to Local Populations

The example database consists of 199 Caucasian and one Asian face. Therefore, depending on where the system is used, the database for importing features must be adapted to the local population. It is not necessary to collect more 3D scans, but photographs of individuals are sufficient. By fitting the morphable model to the faces in the photographs (cf. Section 3.2.4), we obtain 3D faces that can be used to augment the database for feature selection. Even though the reconstructed shapes are in the linear span of the original database, the method produces faithful reconstructions that capture the details of the new faces. In terms of texture, mapping the color values from the images to the model using illumination-corrected texture extraction [BV99] adds new dimensions to the vector space.
Using this technique, we derived three-dimensional head models for the FERET database of photographs [PWHR98] and made them available for feature selection. Figure 5.7 demonstrates that typical features of faces can be transferred to the composite image given only single images of new individuals, even for ethnicities not present in the training database.

5.1.7 Constraints

Sources keep complaining about the database approach to feature selection: either they cannot image what the features would look like in their composite, or their mental image is affected by the many dissimilar faces. Therefore, at any time during the editing process, the user can

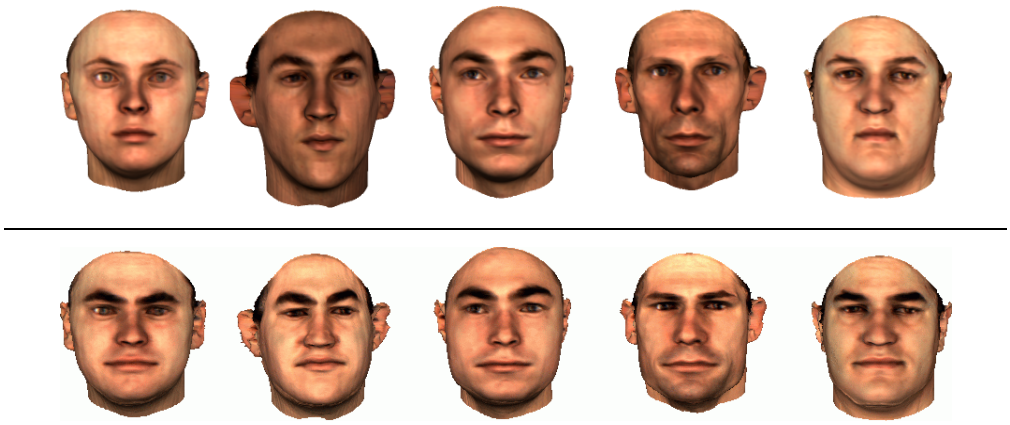


Figure 5.8: **Applying constraints to example faces.** Top row: original faces. Bottom row: transformed faces with common settings for skin color, obesity, mouth width, and eyebrow bushiness.

restrict the database to only display faces of a certain gender, age range, or set of ethnicities (see Section 5.1.1). This has the advantage of showing only those images that fit the characteristics of the target face. The drawback, however, is the reduction in size of the database the user may choose from. It is more efficient here to approach the problem from a different direction and adapt the examples in the database to show the most salient traits of the target face.
To this end, we introduced the possibility to constrain an attribute with respect to shape and texture. Applying a constraint to a database of faces will adapt all faces to show the attribute with the desired intensity. Thus, when the user has set an attribute that is crucial for the appearance of the face, he can store this attribute's value as a constraint and adjust all faces in the database to it. By bringing the examples in the database as close as possible to the source's mental image, the source will be better able to imagine how individual example features will change the appearance of the edit face. Figure 5.8 shows an example where constraints for skin color, obesity, mouth width, and eyebrow bushiness were applied to a set of faces from the example database.

As adumbrated in Section 5.1.4, attribute constraints are also useful to make attribute values about which the user feels confident immune to future modifications by other, correlated attributes. The example from Section 3.2.3 illustrates such a situation: if the user is content with the value for masculine appearance, but then increases the attribute value for the distance between eyes and eyebrows, the result will look less masculine than before due to the fact that gender and eye–eyebrow distance are correlated. Consequently, the user would have to iteratively re-adapt attribute values.
If, however, attribute values of the current edit face are saved as constraints, modifications of these values by subsequent operations are automatically counterbalanced: the system maps the face back to the residual subspace defined by the constraints.
Nature and mode of operation of these constraints are detailed in Section 3.2.3. Figure 5.9 shows the dialog for managing the user-defined constraints.

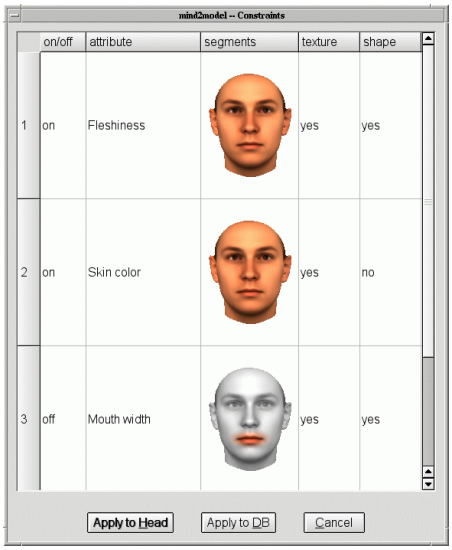


Figure 5.9: **Dialog for managing attribute constraints.** Attribute constraints can be restricted to texture or shape and to sets of segments. They may be applied to the edit head or to the database.

5.1.8 Hair Styles and Crime Scenes

The hair style is one of the first things we notice in a person. As it has a large influence on the way a face is perceived, it is unthinkable that a facial composite system does not offer a selection of hair styles. The face models from the database that were used to learn facial attributes were captured without hair. Thus, we cannot handle hair in the same way as other facial attributes. Despite impressive progress in recent years, 3D hair modeling and rendering is still an open field of research. Even with photorealistic rendering techniques, collecting a database of hair styles would require to re-model existing styles. Therefore, we follow the approach of [BSVS04] described in Section 3.2.5 and render the 3D face model into photographs of hair styles.
This approach means sacrificing arbitrary viewing directions to photorealistic appearance. This can be partially remedied by providing views of the same hair style from different angles. Since the FERET database offers photographs from multiple viewpoints for every face, we decided to use images of people with different hair styles from this collection. After manually selecting 5–15 feature points in each hair style image once, pose, illumination, and contrast for inserting arbitrary target faces are estimated automatically. This estimation works for both color and monochrome photographs. We use the monochrome images from the FERET database, since it was easy to color the hair using image editing software. Thus the user can select a color (light, dark, and red blonde; light, dark, and red brown; black; grey) together with the hair style. Figure 5.10 depicts the hair style dialog of our system.

With the same technique, the composite face can also be rendered into images of crime scenes similar to the relevant incident (as depicted in Figure 5.13). This may help the witness to better recall the situation and remember more detail.

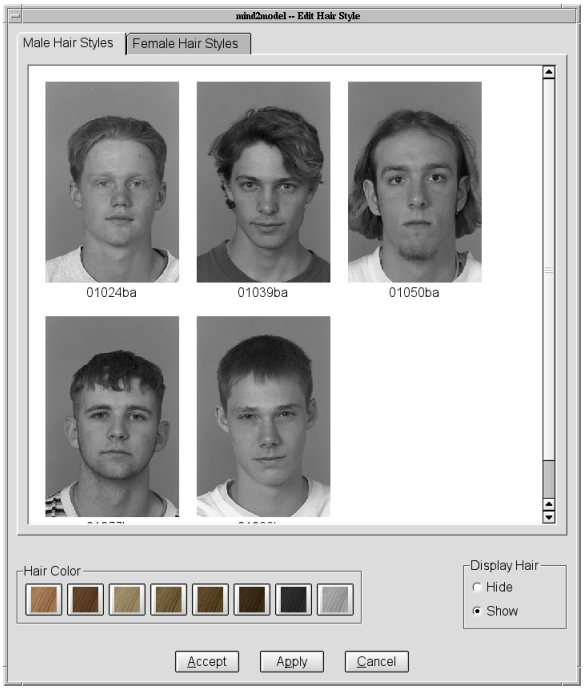


Figure 5.10: **Selecting hair styles.** The user selects hair style and color for the composite face.

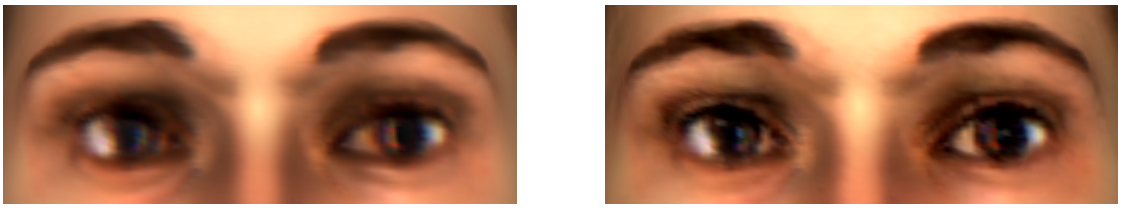


Figure 5.11: **Adding high frequency detail.** Left: eye region of the original composite. Right: eye region of the composite with high frequency detail added. Iris and sclera are better separated, the crease above the eyelid is more prominent, and the transition between eye and surrounding skin is more pronounced.

5.1.9 Adding High Frequency Detail

Due to the limited capabilities of the lens used for capturing the texture during the scanning process, and due to repeated resampling of the data, the reconstructed faces are somewhat blurred, giving the impression that soft-focus was used. In order to make the composite appear sharper, high frequency detail can be added to the face.
Fitting the morphable model to a face in a photograph using illumination-corrected texture extraction (see Section 3.2.4) yields a texture vector with high frequency information. This detail is missing in the texture obtained by scanning the same person. Subtracting this scanned texture from the reconstructed texture gives the desired high frequency detail vector. Since all scans and generated models are aligned, a detail vector from any person can be used. Adding a multiple of such a vector to a face will have an effect especially on eyes and skin structure. Also, the face looks less artificial, as can be seen in the example in Figure 5.11. We found a factor of 0.7 to yield good results. The decision of whether or not to use this feature is up to the user.

source	target	after 1/2 day	after 1 day
S_{f1}	T_m	`mind2model`	PHANTOM
S_{m2}	T_m	PHANTOM	`mind2model`
S_{m1}	T_f	`mind2model`	PHANTOM
S_{f2}	T_f	PHANTOM	`mind2model`

Table 5.3: **Schedule of the experiment.** Each target (T_m male, T_f female) was described by both a female (S_{f1} or S_{f2}) and a male source (S_{m1} or S_{m2}). Every source described the same target twice: once for composite creation with PHANTOM and once with our system. Thereby, succession alternated between sources and was balanced with respect to target person and to source gender. Delays were half a day and one day, respectively.

5.1.10 Implementation Issues

The target face can be saved as a 3D model for further operations. In particular, it is possible to change individual facial attributes later on or to display the face with different rendering parameters. Upon starting the program, the user can either accept the average face as starting point, or load a face from either a previous session or the database of examples.

In order to allow an easy extension of the system, all data are managed through parameter files. If the user wants, for instance, to add an attribute, he needs to rate each example face from the original 3D database, i.e. assign values depending on the saliency of the attribute to the faces. Labeling the entire database of 200 faces takes about 15 min per attribute. From these ratings, the attribute vectors are computed automatically as described in Section 3.2.2. We have found the method to pick up trends in ratings reliably, so labelings may be spontaneous and do not require too much care. If the user now adds the attribute name together with the location where the attribute vector is saved to the appropriate attribute category in the parameter file, the GUI will automatically give access to the new attribute in the respective dialog in the next session.
Similarly, to add a database, the user must create a file containing the locations of all new head files, the subjects' ages, their gender and ethnicity. The extended name of this new file is then added to the parameter file for database management. In the dialog for importing features from example faces, a tab for the new database is created automatically. Augmenting the collection of hair styles works analogously.

5.2 User Study and Results

To evaluate our system, we conducted a user study involving four source persons (S_{f_1} and S_{f_2} female, S_{m_1} and S_{m_2} male, age range 25–60 years) and two target persons (T_m male, T_f female, age range 25–30 years). Sources and targets did not know each other. Each target person was assigned to two source persons. Sources saw "their" target for 60 seconds. After adequate delays (half a day up to one day), all source persons participated in reconstructing the target faces using both `mind2model` and the commercial PHANTOM PROFESSIONALxp© system. Each target was described by both a male and a female source. One source per target started with the PHANTOM program and did their second reconstruction with `mind2model`, while the other source person proceeded vice versa. Again, this was balanced for male and female sources. The overall schedule is shown in Table 5.3.

The police procedure for the creation of composites still involves a forensic artist who operates the composite software at the witness' instructions. We were able to secure the help of a professional forensic artist with considerable experience, Herrn Hans from the Landeskriminalamt (LKA) Saarland[1]. During his career at the LKA, he had already created more than 2000 facial composites. Following his normal routine, he generated each composite image in collaboration with one source using the PHANTOM system and Adobe Photoshop®. First, he asked the source to describe the target person and the circumstances of the encounter in as much detail as possible in order to refresh the witnesses' memory. In addition, the interviewer gets an impression of the location and its particularities as well as of the target person. The actual reconstruction is done with the PHANTOM software. The source isolates from a huge database of color images (roughly 1300 male and 250 female faces) all faces that are in any way similar (nose, eyes, shape, hair style etc.) to the target. The face that best fits the target with respect to face shape, age, skin color etc. is chosen as outline, into which (possibly scaled) features from the rest of the selection are arranged according to the witness' instructions. Apart from affine transformations, the system also offers image editing operations. After the forensic artist had assembled a crude composite, he imported it into Adobe Photoshop® for final retouching. Each composite image was created in about 2–3 hours.
In contrast, our system was operated by computer scientists who naturally lack the forensic professional's psychological background and finesse in conjuring up a detailed image of the target face before the source's mental eye. Following Herrn Hans' example, we also let the witness describe the encounter and the target person prior to the reconstruction. We adapted the average face to this description in order to obtain a rough first approximation. This initial composite was then further manipulated in the presence of the source by translating his / her comments and directions into editing operations. Among these, manipulations of attribute sliders constituted the majority. Since the variety of hair styles in our system was not sufficient for the sources to find a satisfying hair style, we interactively created hair style images according to the sources' descriptions using an image editing program. It took on average 1.5–2 hours to create each target face model.
A comparison of the results created by both our system and the commercial PHANTOM software is shown in Figure 5.12. The overall quality of the results shows that creating recognizable composites is a hard problem for witnesses. In their experiments Laughery and Fowler [LF80] compared the quality of sketches to Identi-Kit composites and concluded that the Identi-Kit technique itself was the main limiting factor. The extensive use of Adobe Photoshop® by the forensic artist excludes this reason. Both for the commercial product operated by the expert and for our system operated by laymen, the main constraint was the sources' limited ability for accurate face recall rather than the flexibility of the system.

Rendering the composite face into an image of a simulated crime scene matching the experience of the witness (as in Figures 5.13) will set the face into proportion to its surroundings and may thus help the source to remember valuable detail.

[1]The LKA is in charge of solving the most serious crimes on state level.

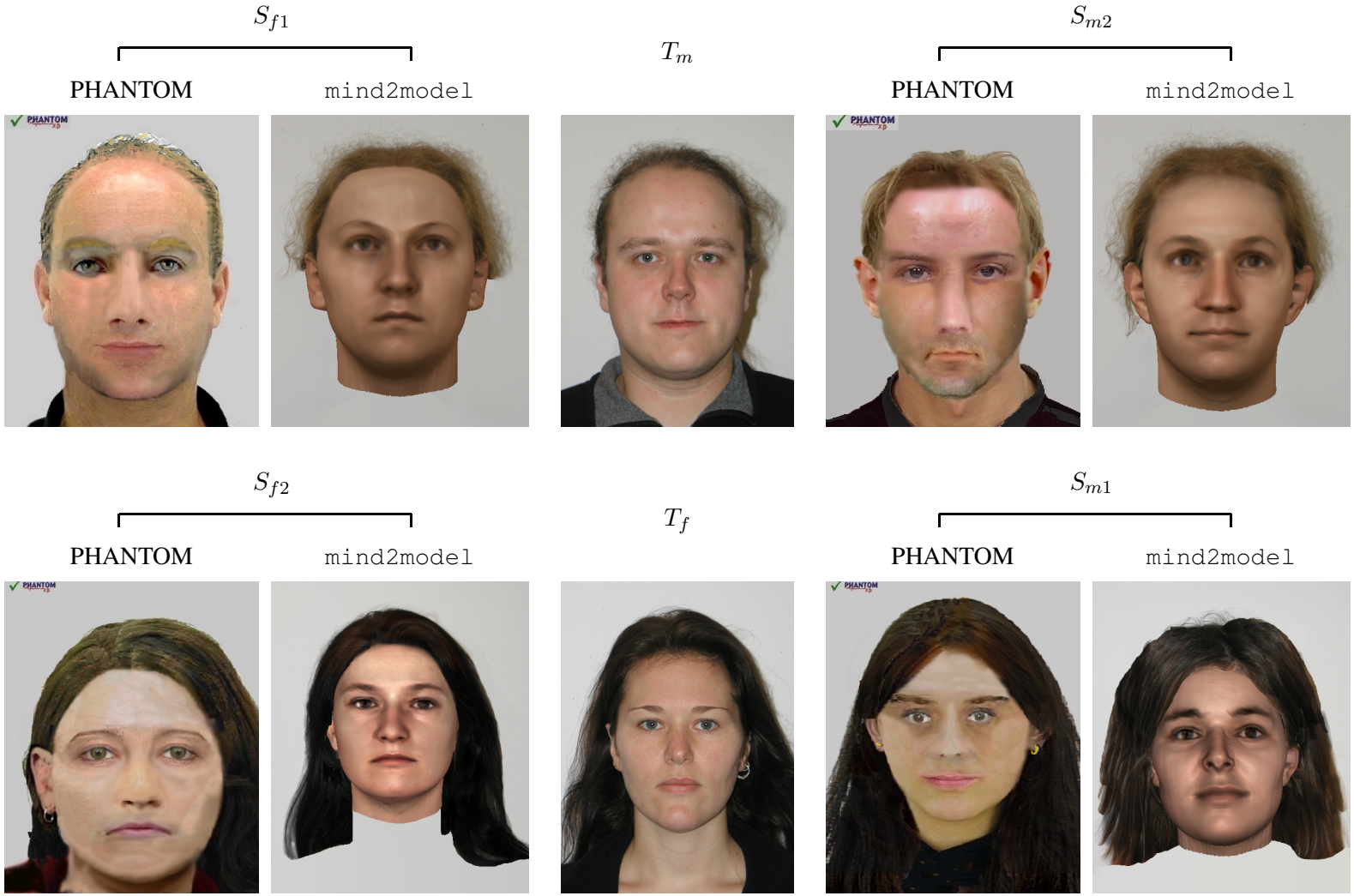


Figure 5.12: **Results from the user study.** Top row: results for the male target T_m. Bottom row: results for the female target T_f. Photographs of the targets are in the center of each row, flanked on the left by the reconstructions according to the female sources S_{f1} for T_m and S_{f2} for T_f, and on the right by those according to the male sources S_{m2} for T_m and S_{m1} for T_f. Each source participated in two reconstruction, once with the PHANTOM PROFESSIONAL© system and once with our system. See Table 5.3 for a schedule of the experiment.

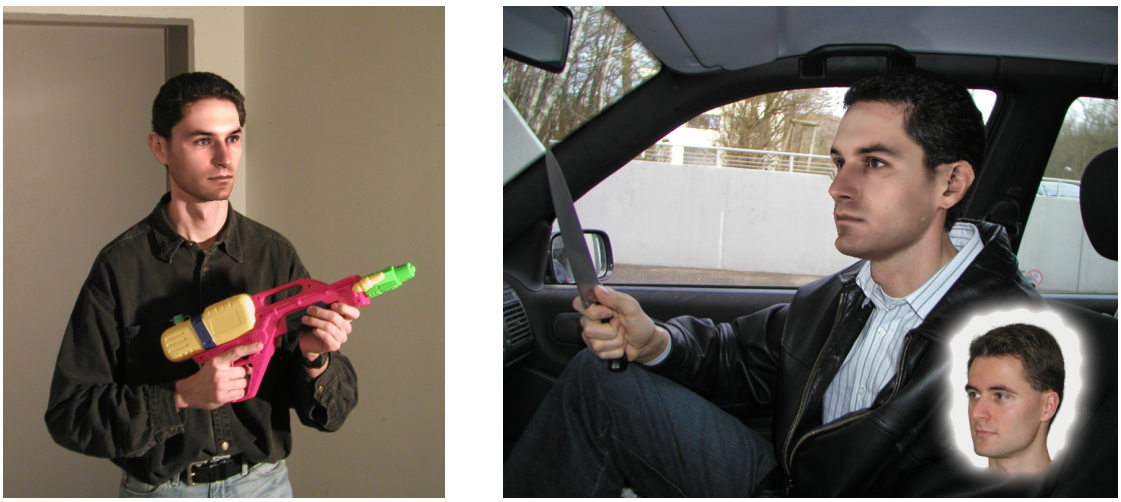


Figure 5.13: **Rendering the composite into a real scene.** Displaying the composite face in a scenario similar to the actual encounter may improve the source's mental image. The inset in the right image shows a photograph of the "perpetrator".

5.2.1 Feedback

Forensic Artist

For both targets, the forensic artist favored one of the composites from our system (for T_m, the reconstruction by S_{f1}, and for T_f, the reconstruction by S_{f2}) and stated that he imagined them to be helpful in a real investigation. Overall, he considered the likeness of T_f by S_{f2} using the `mind2model` program best, except for the missing ear rings. He underlined the importance of such jewelry for identification. In particular, he judged our results as superior in terms of overall face shape, and shape and position of facial features (eyes, nose, mouth). He also appreciated the intuitiveness and continuous variability of the attribute sliders.
In addition, Herr Hans performed an experiment among the colleagues in his department. After first viewing the photographs of the target persons and then the composites, they rated the composites for each target. Table 5.4 shows the results, with the overall ranking in the last row. It confirms the forensic artist's ranking: the general favorites are the composite by S_{f2} with `mind2model` for the female target and the reconstruction by S_{f1} with `mind2model` for the male target.

Sources

One source person favored the PHANTOM software, while the remaining three subjects were more comfortable with the `mind2model` system. One reason for their preference was the fact that the professional composite showed a static front view only, while with our system they were able to view the face from all directions. They used this possibility to model the nose and chin. One rather short subject found it helpful to tilt the model so as to simulate her view of the tall target. Although in the PHANTOM system it is theoretically possible to create an additional 90° view of the face, this requires the database to be present in a second side-face version. Even if this had been the case in our forensic artist's version, it would have been very time-consuming, since all post-processing steps for the front view must be executed analogously for the side view.

judge	**male target** T_m				**female target** T_f			
	best		$\longrightarrow$	worst	best		$\longrightarrow$	worst
J_1	S_{f1} m	S_{f1} P	S_{m2} m	S_{m2} P	S_{f2} m	S_{m1} P	S_{f2} P	S_{m1} m
J_2	S_{f1} m	S_{m2} m	S_{f1} P	S_{m2} P	S_{f2} m	S_{f2} P	S_{m1} P	S_{m1} m
J_3	S_{f1} P	S_{f1} m	S_{m2} m	S_{m2} P	S_{f2} P	S_{f2} m	S_{m1} P	S_{m1} m
J_4	S_{f1} m	S_{m2} m	S_{f1} P	S_{m2} P	S_{f2} m	S_{m1} P	S_{m1} m	S_{f2} P
J_5	S_{m2} m	S_{f1} m	S_{f1} P	S_{m2} P	S_{m1} P	S_{f2} m	S_{f2} P	S_{m1} m
J_6	S_{f1} m	S_{f1} P	S_{m2} m	S_{m2} P	S_{f2} P	S_{m1} P	S_{f2} m	S_{m1} m
J_7	S_{f1} m	S_{f1} P	S_{m2} m	S_{m2} P	S_{f2} m	S_{f2} P	S_{m1} m	S_{m1} P
J_8	S_{f1} m	S_{m2} m	S_{f1} P	S_{m2} P	S_{f2} m	S_{f2} P	S_{m1} P	S_{m1} m
J_9	S_{f1} P	S_{f1} m	S_{m2} P	S_{m2} m	S_{m1} m	S_{m1} P	S_{f2} m	S_{f2} P
J_{10}	S_{m2} m	S_{f1} m	S_{m2} P	S_{f1} P	S_{f2} m	S_{m1} m	S_{f2} P	S_{m1} P
J_{11}	S_{f1} P	S_{m2} m	S_{f1} m	S_{m2} P	S_{f2} m	S_{m1} P	S_{m1} m	S_{f2} P
J_{12}	S_{f1} P	S_{f1} m	S_{m2} m	S_{m2} P	S_{f2} m	S_{m1} m	S_{f2} P	S_{m1} P
J_{13}	S_{f1} m	S_{m2} m	S_{f1} P	S_{m2} P	S_{f2} P	S_{f2} m	S_{m1} m	S_{m1} P
overall ranking	S_{f1} m	S_{f1} P	S_{m2} m	S_{m2} P	S_{f2} m	S_{f2} P	S_{m1} P	S_{m1} m

Table 5.4: **Expert ratings.** All subjects (J_1 to J_{13}) were professionals at the LKA Saarland. After being shown the photographs of the targets, they ordered the composites (see Figure 5.12) according to perceived quality. The last row shows the overall ranking. P and m are abbreviations for PHANTOM and `mind2model`, respectively, and S_{f1}, S_{f2}, S_{m1}, and S_{m2} stand for the source persons, i.e. S_{f1} m in column 2, for example, is a composite of target T_m by S_{f1} using `mind2model`. Source: Herr Hans, LKA Saarland.

With the PHANTOM system, the sources also missed the possibility to replace facial features during the post-processing without loosing all intermediate editing steps. Some subjects had problems with the database approach in general. One source complained that "*having seen so many other faces* [in the database]*, I can't recall what's right or wrong anymore.*" This is in accordance with the findings of Deffenbacher et al. [DCL81], who report that memory for faces suffers greatly from retroactive interference, i.e. forgetting due to exposure to many different faces. This seems to be especially noticeable in the short term. Davies and Christie [DC82], on the other hand, found that interference is not a major problem.

Concerning the reconstruction process with PHANTOM, another subject said: "*Despite the huge amount of faces in the database, I wasn't able to find the right eyes: I couldn't imagine what all those eyes would look like in the face I was going to describe.*" We found indeed that most of the statements of sources were of the type "*make the nose wider*", so the slider-interface turned out to be more appropriate for the reconstruction process than the standard database selection of the professional systems that our software also offers.

Another step we took to avoid contaminating the source's mental image was to modify the average face to roughly fit the source's descriptions before the subject saw it for the first time. The source avoided looking at the screen while we started the program and modified the average face to comply with the initial description. Only then did the source join in the composite creation in the usual way by giving instructions to the operators. Hence, sources did not start from the aver-

age face, but from a first approximation of their mental images. Hereby we found that importing features from the database for this first estimate seemed to confuse and hence frustrate subjects, probably because the features were too specific. When we used attributes alone, sources were rather content with the first guess. One even commented that it looked already more like the face she had in mind than the end result of the composite created using the commercial product. This was certainly a gross exaggeration, especially when considering the skills of the forensic artist, but shows the amount of frustration the unnaturalness of a photofit face can induce.

5.3 Conclusions

Creating recognizable facial composites is a difficult problem due to the inadequacy of the human brain with respect to face recall. The main advantage of the `mind2model` system is that due to the underlying statistical model it supports the user in the difficult task of bringing his mental image to the screen without restraining him. For overall editing operations, the program presents the most probable solution while leaving the user the freedom to override this result. The system is intuitive to control and fine-grained. The user always works with an entire face and hence need not consider isolated facial features. Viewing the composite face model from several directions is helpful when defining the overall face shape, when editing silhouette features, or to simulate a different perspective. By definition of the morphable model, face models of a single target created from several sources can be morphed easily to obtain weighted combinations.

Psychological studies indicate that human memory for faces is hurt through retroactive interference if subjects are exposed to images of different faces between the study and test phase [DCL81]. Therefore, we reduce the variety of faces our sources are exposed to by adjusting the example faces to meet user-specified constraints. Moreover, wrong clues about features of the target face interfere with the sources' recollection significantly [JD85]. Our technique accounts for this by showing at any time the most plausible face according to the correlations estimated from the database.
Our control experiment with an experienced professional who was operating commercial software shows that, apparently, the problem of reconstructing faces is intrinsically difficult. The sources had seen the unknown targets for only 60 seconds and described them half a day or one day later. The comparison of the results indicates that the limiting factor is the deficiency of human consciousness with respect to exact recollection of faces, not the accuracy of our system. This is also supported by the fact that composites created from photographs using our system (Figure 5.14) are a lot more accurate than those created from memory.
Considering the forensic artist's and the sources' positive reactions, we may conclude that we took a step into the right direction towards our goal of achieving better facial composites than existing systems in a less tedious process.

For adding details such as birth marks, scars, or jewelry, we currently propose to follow the forensic artist's approach and finish an image of the completed head in the desired view using an image editing program. Of course it would be desirable to have databases of such accessories and facial particulars as well as beards and hair styles in 3D, in order to be able to add such detail at any stage of the creation process, and to allow views of the final composite head from any direction.

Figure 5.14: **Composite creation from photographs.** Left: composite (by a non-artist). Right: original photograph. The result from this ideal situation demonstrates the flexibility and accurateness of `mind2model`.

Aging is an important feature in a composite system. It allows to update composites years after the crime. By collecting facial scans from various age groups, age-related shape and texture properties could be learned and applied to the composite face.
Using a similar approach to the one presented here, one could generate photospreads for identification of suspects by witnesses. Typically, a photospread consists of a picture of the suspect and five to nine more or less similar "fillers". These fillers could be generated by defining constraints for the most prominent features of the suspect, such as nose curvature for a very hooked nose, and varying the rest of the face by constrained random permutation. Hereby one must carefully avoid that the original image stands out. An interesting question would be how to determine the right amount of perceptual similarity.
The underlying morphable model lends itself also to face recognition [BV03]. This means that mug-shot database search can be done using composite faces created with `mind2model`. Usually, if the police suspects that the perpetrator already is in their mug-shot database, the witness is asked to go through the (usually huge) database to try to identify the criminal. This can be facilitated by first creating a composite with `mind2model` according to the description by the witness, and then running an automatic search on the database with the composite as query. Now the witness only has to look at the set of best hits returned by the search instead of at the entire database. Not only should that speed up the process, but it also reduces the stress on the witness and avoids exposure to too many "wrong" faces.

6

Hands

Our hands play a vital role in every aspect of our daily lives. Humans use their hands for communicating, for eating, playing, writing, working, in a nutshell: for everything. Most people take the effectiveness and dexterity of their hands for granted without being aware of their complicated structure and the high level of optimization. However, there is more than the mechanical perfection to our hands: stretching from spiritual significance (e.g. blessing, palm reading), over idiomatic expressions (e.g. "to put one's life in someone's hands"), to the act of shaking hands, not only for greeting but also for expressing feelings like gratefulness or sympathy, the central importance of hands is mirrored in a broad spectrum of symbolism.
Cultural differences exist in all areas, including gestures. They do not only lie in the amount of gesturing, but also in the type of gestures. An often-cited example is a former mayor of New York with an Italo-Jewish background. Notably, he could switch gesture language along with spoken language seamlessly.
In spite of the ubiquity of hands in daily life, but probably due to their immense complexity, hands have only very recently begun to receive due attention in computer graphics. Although the number of possible applications is large, only a handful of sophisticated hand models have been developed to date, and even less such models existed at the time we conducted the research described here. Virtual hand models can be used for teaching and practicing sign language, and for visualizing translations from speech or text into sign language. They come in handy for teaching other manual skills as well, for instance operating machines, and for giving online usage or assembly instructions. In immersive environments, hand models are required in the simulation of the haptic dimension: for manipulating a virtual object, visual feedback is helpful. Close-ups in CG movies and games ask for natural models with a lot of detail and convincing movements. Such situations include communicative hand gestures, involuntary twitches of the hand that betray the character's true feelings or intentions, romantic scenes, and tool manipulation. High demands arise also from the medical field. In systems for hand surgery planning, a maximum of functionality of the hand must be provided to aid the surgeon in his decisions.

Incited by this abundance of exciting areas of application that can absolutely compete with those of the (at least in computer science) much more thoroughly investigated face, we created a system for hand modeling and animation as described below.

This chapter is organized as follows: after an introduction to the anatomy of the human hand in Section 6.1, the above mentioned model is described in more detail (Section 6.2). Results are presented in Section 6.2.4, succeeded by a final discussion in Section 6.2.5. In Section 6.3, we present an application of the hand model: a baseball pitcher's hand is visualized during ball

release for different pitches. The hand pose data was obtained from tracking the hand motion of a real pitcher.

6.1 Anatomy of the Human Hand

To be able to execute powerful movements and fine motor manipulations alike, the anatomy of the human hand has evolved to a high level of complexity. A multitude of small parts work together to perform the diverse tasks of a hand through their concerted efforts. In order to realistically model and animate this intricate body part, a thorough insight into the structure and functioning of the individual building blocks is required. With respect to literature, this part relies mainly on [BH99, PP01, Fun93, Bau87].
Structure and joint hierarchy of the hand are determined by the underlying skeleton, which is described in Section 6.1.1. Section 6.1.2 gives an account of the degrees of freedom at individual joints as well as of joint structure in general. A description of the motor of the hand, i.e. its muscle apparatus, follows in Section 6.1.3, considering the layout of the musculature both at the microscopic and at the macroscopic level. The topic is closed with a brief discussion of the texture and properties of human skin in Section 6.1.4.

6.1.1 Skeleton

The hand's rather small volume contains 27 bones (see Figure 6.1), not counting *radius* and *ulna*, which lie in the forearm. Together, *scaphoid*, *lunate*, *triquetrum*, *pisiform*, *hamate*, *capitate*, *trapezoid*, and *trapezium* constitute the *carpal* bones. They connect to *radius* and *ulna* through the wrist joint on their proximal side, and via the *carpometacarpal* joint to the *metacarpals* distally. Although the *metacarpals* lie completely within the palm, they belong to the fingers. The actual fingers are supported by the *phalanges*. The thumb has only two phalangeal bones, the first *proximal phalanx* and the first *distal phalanx*, while the remaining fingers also have *middle phalanges*. Tiny *sesamoid* bones are located both on the radial and on the ulnar sides of the first metacarpal next to the *metacarpophalangeal* joint. They are embedded in the tendons of the *adductor pollicis* and the *flexor pollicis brevis* muscles. In the fingers, more sesamoid bones may be present, varying between individuals. Sesamoids serve to increase the muscles' mechanical advantage at the joint and to reduce pressure on the underlying tissue. The largest human sesamoid bone is the *patella* in the knee.

6.1.2 Joints

The resulting number of degrees of freedom (DOFs) is enough to make any animator sweat. The *proximal interphalangeal* (PIP) and *distal interphalangeal* (DIP) joints of the fingers and the *interphalangeal* (IP) joint of the thumb have one DOF each for flexion / extension. The MCP joints of the fingers have a second DOF for adduction (towards the middle finger) and abduction (away from the middle finger). This second axis is not at a right angle to the fully extended finger, but rather rotated by approximately -30° around the flexion / extension axis. Consequently, the rotation volume of the *proximal phalanx* during adduction / abduction is a disc segment only if the finger is extended backwards by 30°, because then the *phalanx* is perpendicular to both axes. Otherwise, the rotational solid is a cone. The cone's diameter diminishes with increasing flexion until the finger is parallel to the abduction / adduction axis, i.e. merely revolves around itself [BH99]. Figure 6.2 compares the cone to an umbrella

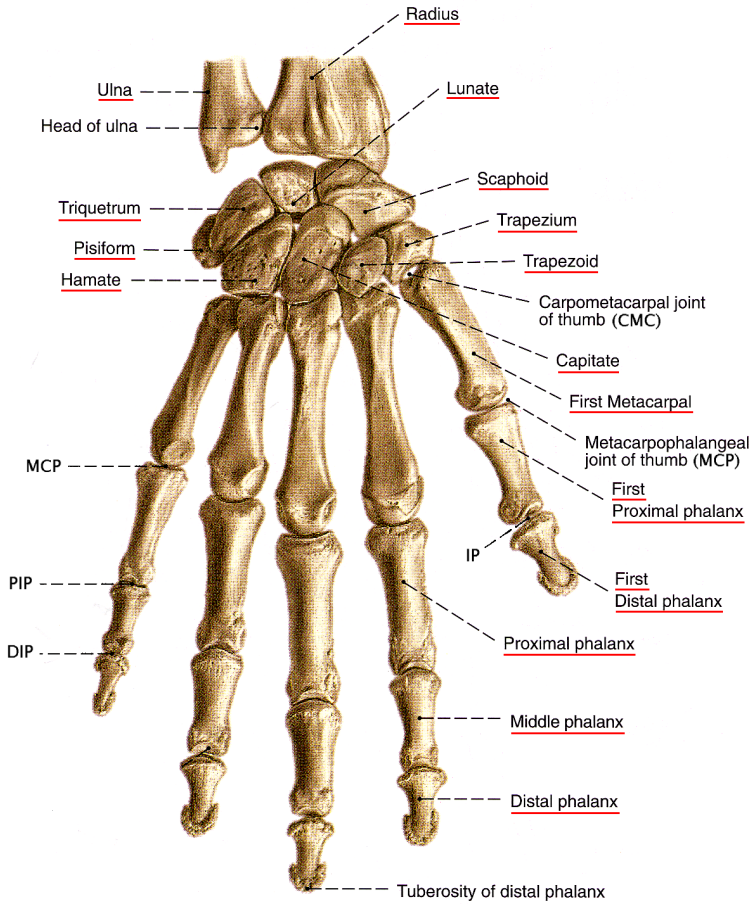


Figure 6.1: **Bones of the human hand and forearm.** Bone names are underlined in red. The *metacarpal*, *proximal phalanx*, and *distal phalanx* bones exist in each finger of the human hand, while the *middle phalanx* bones exist in all fingers but the thumb. Abbreviations: *interphalangeal joint*: IP, *proximal interphalangeal joint*: PIP, *distal interphalangeal joint*: DIP. Source: [PP01].

with its tip at the intersection of flexion / extension and adduction / abduction axis, its handle along the adduction / abduction axis, and its opening angle determined by the flexion angle of the finger. During adduction and abduction, the *proximal phalanx* follows the fabric of the umbrella. As the opening of the umbrella decreases when flexion increases, the diameter of the traced cone is also reduced. When the umbrella is fully closed, the fabric and hence the outstretched finger is parallel to the handle, i.e. to the adduction / abduction axis. As a result, the finger can only rotate around itself. Therefore, if axes perpendicular to the *proximal phalanx* are used to model the joint DOFs, three axes are required to capture the entire range of movement at the MCP joints. Likewise, the *carpometacarpal* (CMC) joint of the thumb is sometimes said to have three degrees of motion. Here, the impression of rotation around a third axis is evoked by the fact that the two real axes are not completely perpendicular. In addition, they do not intersect. The same holds for the thumb *metacarpophalangeal* (MCP) joint (see [BH99]), but its range of motion for adduction / abduction is small. Of the fingers, only the fifth and the ring finger have a movable CMC joint. Rotation angles are very small. The *midcarpal* joint runs between the *scaphoid*, *lunate* and *triquetrum* on the proximal

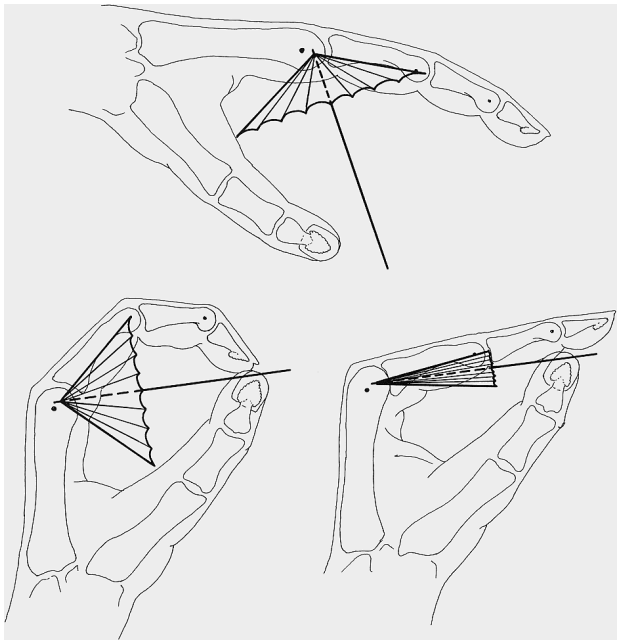


Figure 6.2: **Adduction / abduction movement at the finger MCP joints.** The adduction / abduction axis of the finger is not perpendicular to the outstretched finger, but rotated by about -30° around the flexion / extension axis. Unless bent backwards by 30°, the *proximal phalanx* traces a cone, whose diameter depends on the flexion angle. This varying cone can be compared to an umbrella that is being closed by flexing the finger: its opening diameter will continually decrease until the finger is parallel to the umbrella handle, i.e. the adduction / abduction axis. In that case, the finger is only able to rotate around its long symmetry axis. See also Section 6.1.2. Source: [BH99].

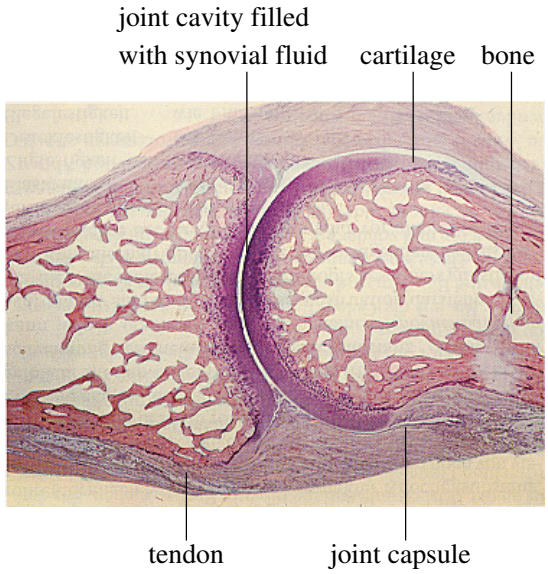


Figure 6.3: **Human finger joint.** Micro-photograph. Source: [Bau87].

side, and the distal carpal row (*trapezium*, *trapezoid*, *capitate*, *hamate*). It permits flexion and extension (together with the wrist joint), and extremely limited rotation. The amount of movement in the joints between the carpal bones (*intercarpal* joints) of the same row is minimal.

Structurally, every joint is enveloped by a *joint capsule*. Underneath the capsule, ligament connects the two bones. The bone closer to the center of the body is called *proximal* bone, the other one is called *distal* bone. Each joint face is padded with cartilage. The faces do not touch directly, but between them build the *joint cavity*. It is filled with *synovial fluid*, an extremely effective lubricant, which allows the two cartilage surfaces to glide past each other almost without friction. Figure 6.3 shows a micro-photograph of a cross-section of a human finger joint.

6.1.3 Muscles

The number of muscles is even more abundant than the number of joints or DOFs. Figures 6.4 and 6.5 show the muscles of the palm. For the sake of visibility, several superficial muscles are peeled back. The muscles of the back of the hand are depicted in Figure 6.6. Noticeably, most muscle bellies are located in the forearm. This is apparent for the *extensor digitorum* muscles in Figure 6.6, for instance, where the tendons crossing the wrist are visible, while the muscle bellies are cut off. Consequently, the majority of muscles affecting the fingers also have an effect on the wrist joint. Where a tendon crosses a joint, ligaments keep the tendon close to the joint to prevent bowstringing or exertion of too much leverage.

Muscles are classified according to the location of their muscle bellies. Hence a lot of muscles that mainly affect the hand are counted as muscles of the forearm. In the following, a brief account of all muscles involved in hand movement is given.
The *superficial ventral* muscles of the forearm (i.e. the superficial muscles located on the inside of the forearm) include the *flexor carpi radialis*, which causes palmar and radial flexion of the wrist and carpal joints, and the *palmaris longus*, which involves palmar flexion of wrist and carpal joints and tension of the tendon-like *aponeurosis*. Contraction of the *flexor digitorum superficialis* leads to palmar and ulnar flexion of wrist and carpal joints as well as to flexion and adduction of the MCP joints II-V and to flexion of the corresponding PIP joints. The *flexor carpi ulnaris* causes palmar and ulnar flexion of the wrist and carpal joints.
The *deeper ventral* layer contains only the *flexor digitorum profundus* and the *flexor pollicis longus*, which both cause palmar flexion of the wrist and carpal joints. In addition, the *flexor digitorum profundus* flexes and adducts the fingers at their MCP joints, and also flexes them at the PIP joints, whereas the *flexor pollicis longus* causes adduction and opposition at the thumb CMC as well as flexion at the remaining two thumb joints.
There is only a single hand muscle among the *radial* muscles of the forearm, i.e. the muscles located on the same side of the arm as the thumb. This is the *extensor carpi radialis brevis*, a wrist extensor which also causes abduction.
All *superficial dorsal* muscles of the forearm are concerned with extension: the *extensor carpi ulnaris* restricts itself to the wrist and carpal joints, where it also causes adduction, while the *extensor digitorum* extends all joints from the wrist to the DIP joints of the fingers. At the wrist and carpal joints, it also effects adduction. The *extensor digiti minimi* does the same, but among the fingers only affects the pinky.
The *deep dorsal* layer contains more extensors: the *extensor pollicis longus* (extension of wrist and carpal joints, adduction and reposition at the CMC I joint, and extension of the MCP I

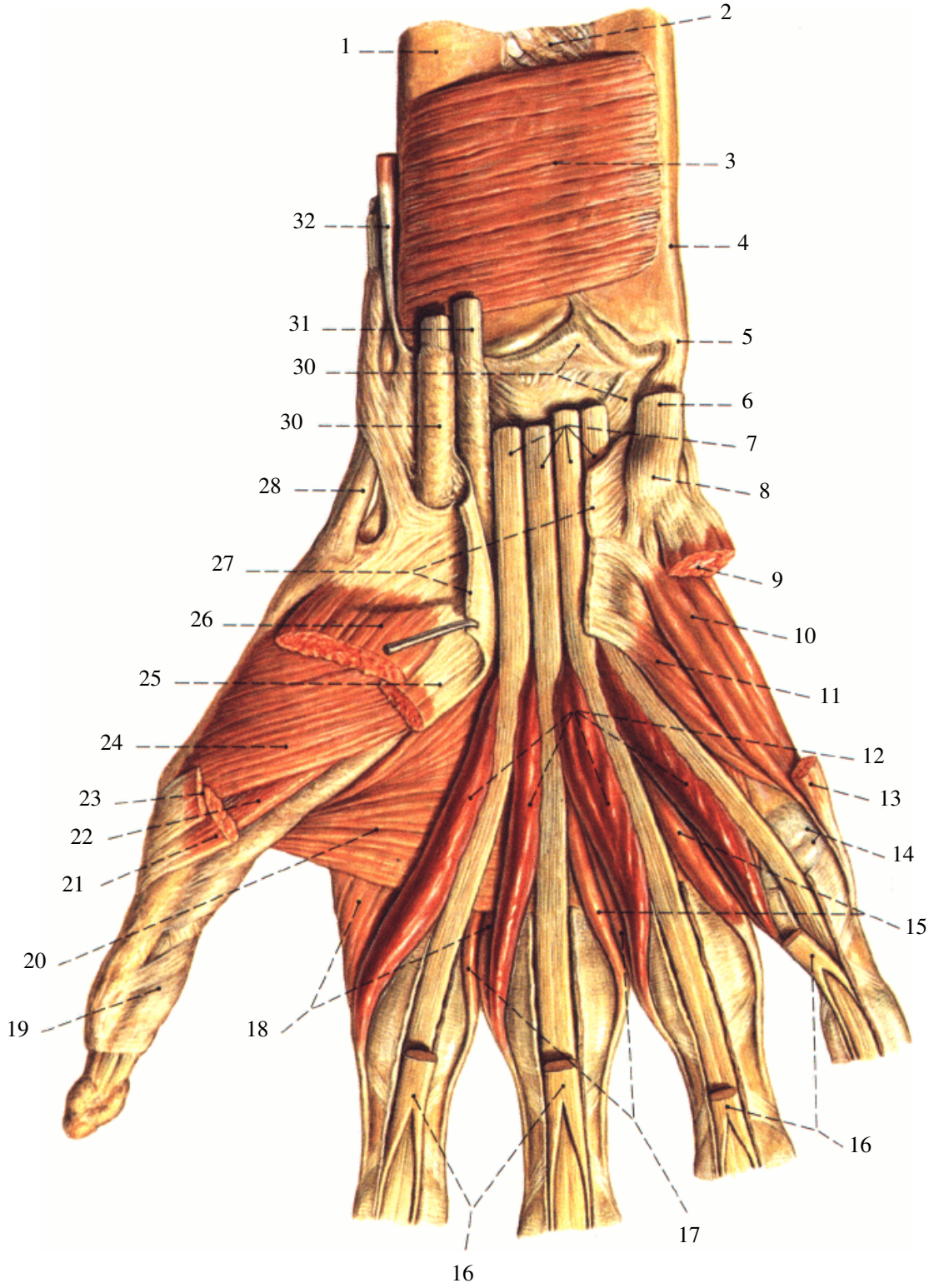


Figure 6.4: **Palmar aspect of hand muscles.** Right hand. For visualization purposes, several superficial muscles have been removed. Numbers refer to Table 6.1. Source: [PP01].

1	radius
2	interosseus membrane of forearm
3	pronatur quadratus
4	ulna
5	ulnar styloid process
6	flexor carpi ulnaris, tendon
7	flexor digitorum profundus, tendons
8	pisiform
9	abductor digiti minimi
10	flexor digiti minimi brevis
11	opponens digiti minimi
12	lumbricals
13	abductor digiti minimi
14	5th metacarpophalangeal joint: joint capsule
15	dorsal interosseus muscles
16	flexor digitorum superficialis, tendons
17	palmar interosseus muscles
18	dorsal interosseus muscles
19	tendinous sheath of flexor pollicis longus
20	adductor pollicis, transverse head
21	flexor pollicis brevis, superficial head
22	flexor pollicis brevis, deep head
23	abductor pollicis brevis
24	opponens pollicis
25	flexor pollicis brevis, superficial head
26	abductor pollicis brevis
27	flexor retinaculum
28	tendinous sheaths of abductor longus and extensor pollicis brevis
29	tendinous sheath of flexor carpi radialis
30	palmar radiocarpal ligament
31	flexor pollicis longus, tendon
32	brachioradialis, tendon

Table 6.1: **Palmar hand muscles.** Numbers refer to Figure 6.4.

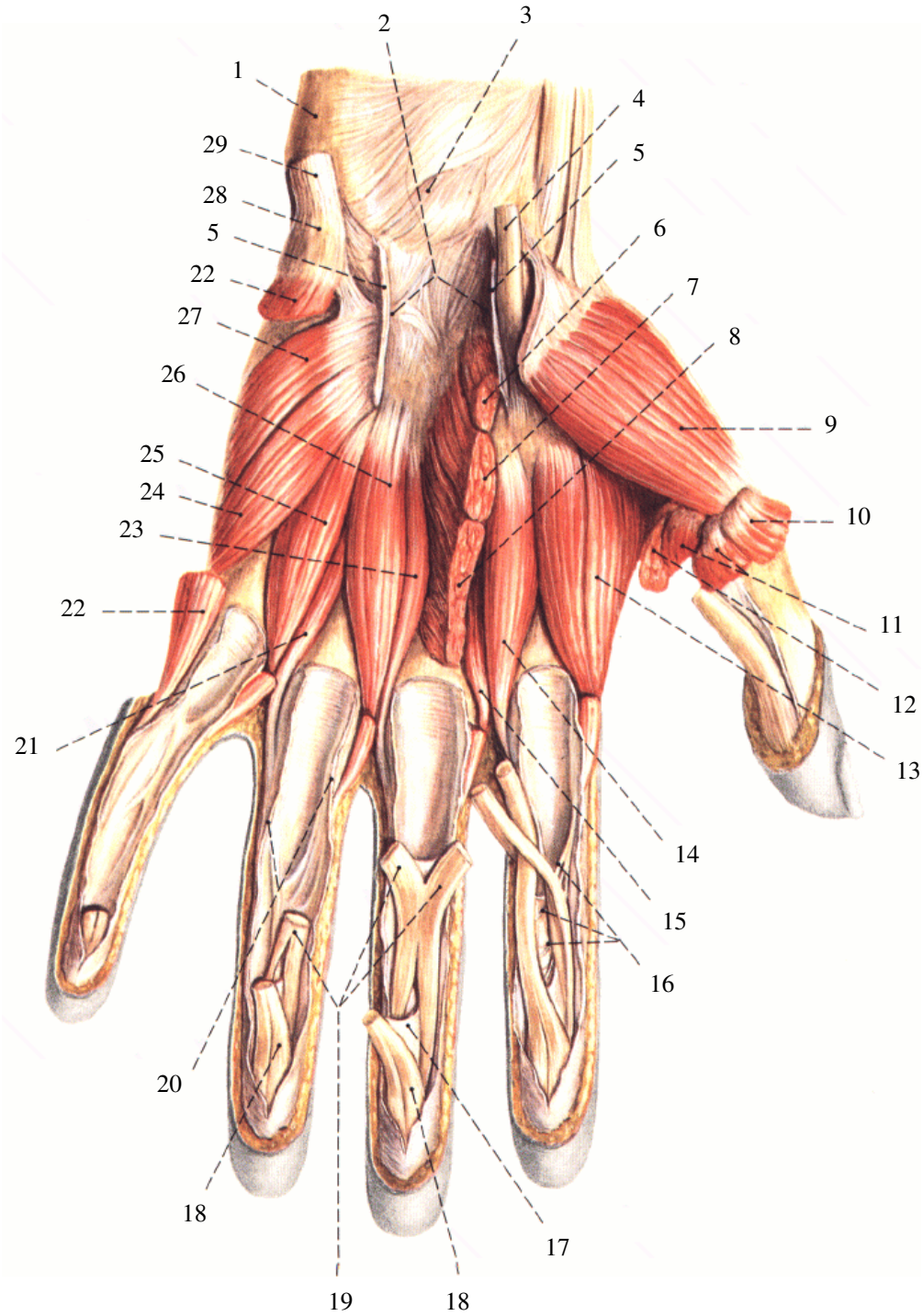


Figure 6.5: **Palmar aspect of hand muscles, deep layer.** Left hand. For visualization purposes, several superficial and the deep *flexor digitorum* muscles have been removed. Numbers refer to Table 6.2. Source: [PP01].

1	ulna
2	carpal tunnel
3	palmar radiocarpal ligament
4	flexor carpi radialis, tendon
5	flexor retinaculum
6	flexor pollicis brevis, deep head
7	adductor pollicis, oblique head
8	adductor pollicis, transverse head
9	opponens pollicis
10	abductor pollicis brevis
11	flexor pollicis brevis
12	adductor pollicis
13	1st dorsal interosseus
14	1st palmar interosseus
15	2nd dorsal interosseus
16	vincula longa
17	vinculum breve
18	flexor digitorum profundus, tendons
19	flexor digitorum superficialis, tendons
20	fibrous and synovial sheath of digits of hand
21	4th dorsal interosseus
22	abductor digiti minimi
23	3rd dorsal interosseus
24	flexor digiti minimi brevis
25	3rd palmar interosseus
26	2nd palmar interosseus
27	opponens digiti minimi
28	pisiform
29	flexor carpi ulnaris, tendon

Table 6.2: **Palmar hand muscles, deep layer.** Numbers refer to Figure 6.5. A *retinaculum* is a band of connective tissue that holds an anatomical building block in place.

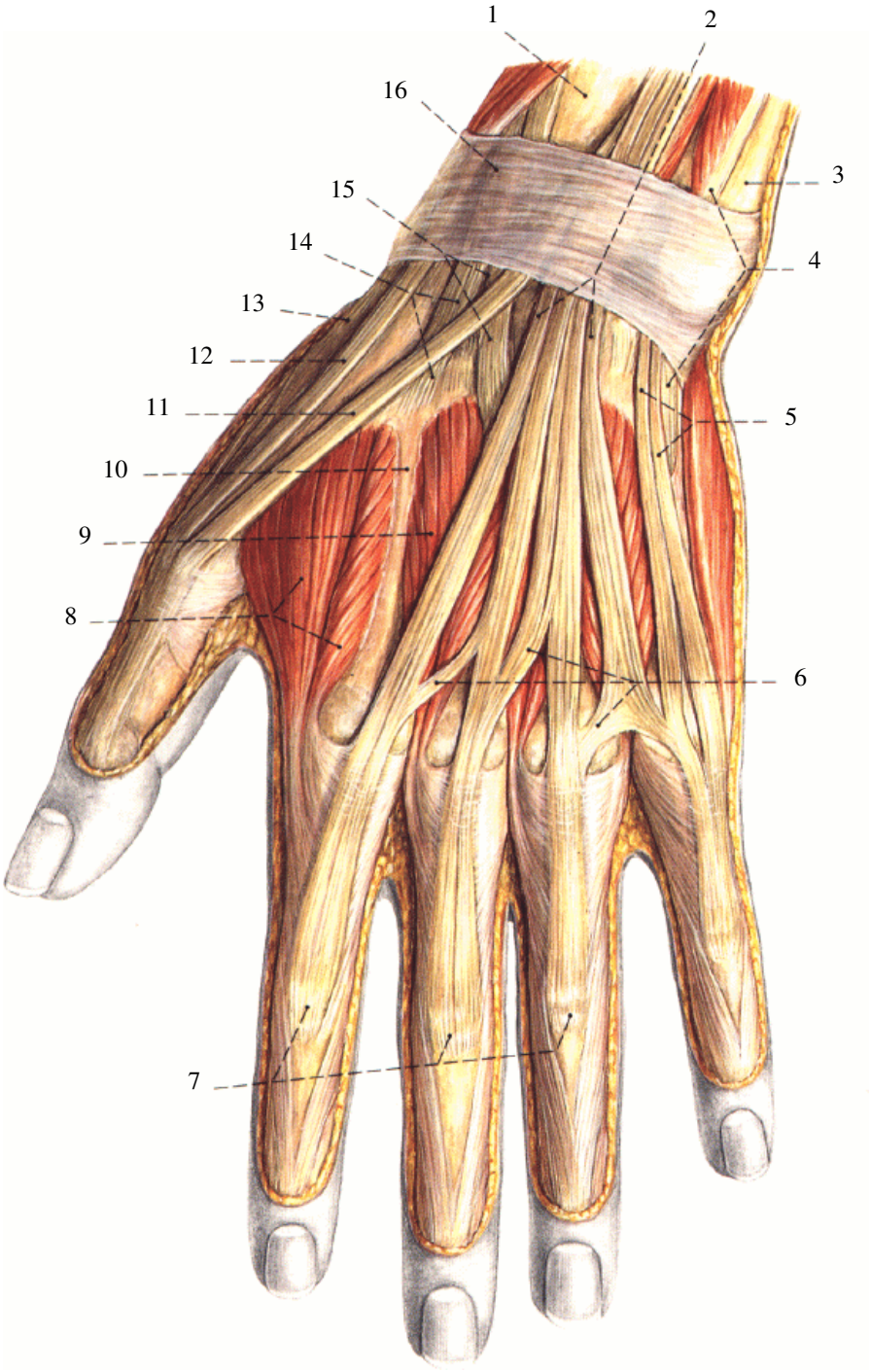


Figure 6.6: **Dorsal aspect of hand muscles.** Left hand. Numbers refer to Table 6.3. Source: [PP01].

1	radius
2	extensor digitorum, tendons
3	head of ulna
4	extensor carpi ulnaris, tendons
5	extensor digiti minimi
6	intertendinous connections
7	interphalangeal joints
8	1st dorsal interosseus
9	2nd dorsal interosseus
10	2nd metacarpal
11	extensor pollicis longus, tendon
12	extensor pollicis brevis, tendon
13	trapezium
14	extensor carpi radialis longus, tendon
15	extensor carpi radialis brevis, tendon
16	extensor retinaculum

Table 6.3: **Dorsal hand muscles.** Numbers refer to Figure 6.6.

and IP joints) and the *extensor pollicis brevis* (flexion and abduction of wrist and carpals, abduction and reposition at the thumb CMC, and extension at the MCP joint) for the thumb, and the *extensor indicis* for the index finger (flexion and abduction at the wrist and carpal joints, extension at the MCP, PIP, and DIP joints, and adduction at the MCP). In addition, the *abductor pollicis longus* causes flexion and abduction at the wrist and carpal joints as well as flexion of the thumb metacarpal.

The *thenar* muscles are concerned with movement of the thumb. Their muscle bellies form part of the *thenar eminence*, i.e. the fleshy prominence at the base of the thumb. The *abductor pollicis brevis* flexes the thumb metacarpal and effects abduction and opposition at the CMC joint. Opposition describes the movement that brings the thumb and the fifth finger in a position where their tips touch. Other muscles concerned with thumb opposition are the *opponens pollicis*, the *adductor pollicis*, and the *flexor pollicis brevis*. In addition, these muscles adduct the thumb at the CMC joint. The *flexor pollicis brevis* and the *adductor pollicis* also cause flexion at the MCP joint.
Opposite to the thenar eminence, in the *hypothenar eminence*, lie the muscles that affect the pinky only. They all cause opposition at the CMC V joint. For the *opponens digiti minimi*, this is the only task, while the *abductor digiti minimi* also causes abduction at the MCP joint and extension at the interphalangeal joints, and the *flexor digiti minimi brevis* abducts and flexes the fifth finger at its MCP. In addition to the muscles effectuating finger motion, there is also the *palmaris brevis*, a superficial muscle beneath the skin that tenses the skin of the hypothenar eminence.
The remaining intrinsic muscles are the four *lumbricals* and the seven *interossei*. The *lumbricals* originate from the tendons of the *flexor digitorum profundus* and insert into the distal phalanges II-V. They cause flexion and radial abduction at the MCP joints II-V and extension at

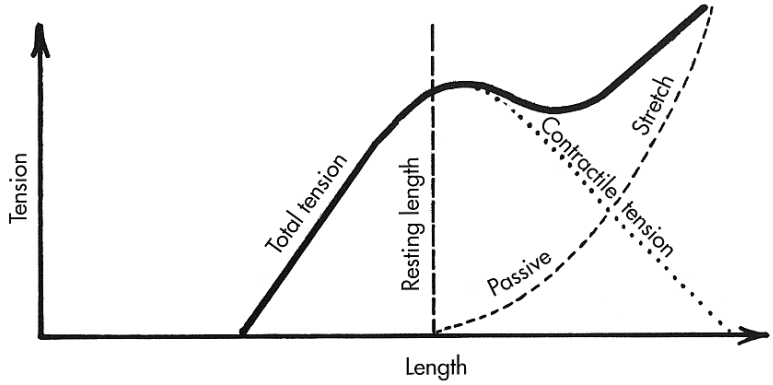


Figure 6.7: **Blix curve.** The length-tension curve after Magnus Blix combines tension from active contraction and from passive stretch. Source: [BH99].

the interphalangeal joints. The three *palmar interossei* affect the index, ring, and little finger, effecting flexion and adduction towards the middle finger at the MCP joints and extension at the interphalangeal joints. The four *dorsal interossei* flex the second to fourth finger at the MCP joints and abduct the index and ring fingers from the middle finger. The latter is rotated radially by the second and towards the ulna by the third muscle.

Considering the mechanical properties of muscles, their output is tension, from both active contraction and passive recoil. The passive component is due to the elasticity of collagen and other tissue that is stretched together with the muscle. The curve for passive lengthening of the muscle in the length-tension diagram in Figure 6.7 shows tension rising continuously as the muscle is being stretched. When the muscle contracts actively, there is an optimal length where the muscle can exert its maximum active contractile force. This state of maximal force coincides with the muscle being at its *resting length*, i.e. the length the muscle assumes when the limb is relaxed.

The muscles in the hand are *striated muscles*. Striated muscles (see Figure 6.8) of mammals have the *sarcomere* as minimal active unit. All sarcomeres are of the same size and condition, so variations in length or thickness between muscles are due to the number and arrangement of the sarcomeres. A sarcomere comprises two parallel *zwischenscheiben* or *Z plates*. On the inner faces of the Z plates, *actin* filaments are attached that interdigitate with *myosin* filaments. This is depicted in Figure 6.8 (right (a)). When the sarcomere is activated, the attraction between actin and myosin increases, resulting in an increase in the fibers' overlap, see Figure 6.8 (right (b)).
Muscle fibers (see Figure 6.9) are composed of several long chains of sarcomeres that are arranged Z plate to Z plate. These chains are called *myofibrils*. Hence, when the sarcomeres contract, so does the muscle fibril. Sarcomeres of adjacent myofibrils are aligned. Muscle fibers are pooled into *muscle fiber bundles* by sheaths of *connective tissue*, which also fills in the gaps between the individual fibers. These bundles finally make up the *muscle*. So muscle tension is the result of the individual sarcomeres of a muscle contracting in unison.
Differences between individual muscles result from differences in fiber length and from differences in the cross-sectional area of the muscle. Fiber length is proportional to potential excursion, while the cross sectional area determines the muscle's maximum possible tension.
Quite often the tendons run along part of the muscle, with fibers originating and inserting along the way as depicted in Figure 6.10. This means that muscle length does not necessarily equal the

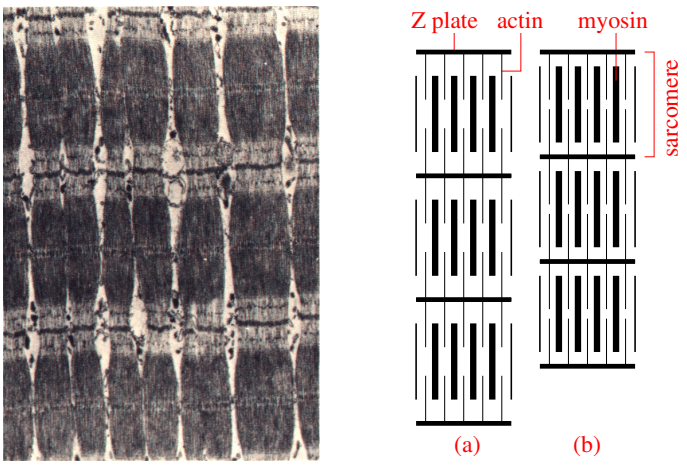


Figure 6.8: **Striated musculature.** Left: each of the eight vertical myofibrils contains three sarcomeres. Source: [Bau87]. Right: (a) schematic view; (b) for contraction, the actin filaments are pulled between the myosin filaments.

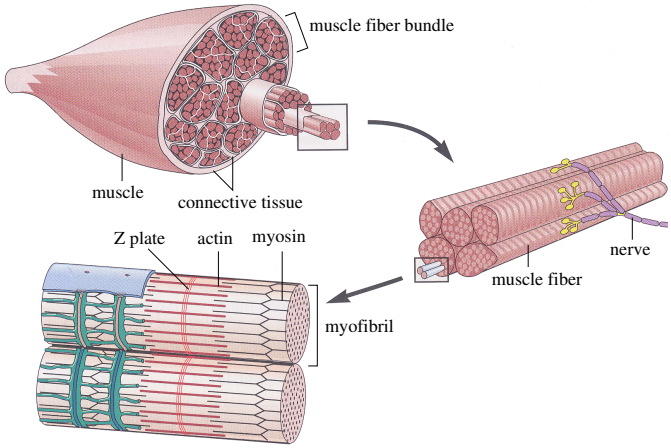


Figure 6.9: **Building blocks of a muscle.** Striated muscles consist of bundles of muscle fibers, with individual fibers composed of long chains of sarcomeres called myofibrils. Source: [Spe98].

length of the muscle fibers. The *flexor carpi ulnaris*, for example, is a very strong muscle with little potential for excursion: the muscle belly may be 25 cm long, but be accompanied during 21 cm by the tendon of insertion, and have fibers of only 4 cm length [BH99].

6.1.4 Skin

Skin consists of three main layers, the *epidermis*, the *dermis*, and the *hypodermis*. Figure 6.11 shows a cross section of human skin.

The *epidermis* unites the surface layers, which protect the body against harmful external influences, such as dehydration and ultraviolet rays. Finger and toe nails originate from this layer. It contains nerve endings, but no blood vessels. At its lowest level, new cells form continuously. They are pushed towards the surface, hornificate, die, and finally scale of. The living, lowest layer is nourished by the *dermis*, a dense ply built of fibro-elastic connective

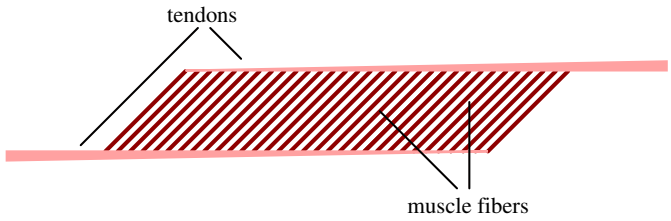
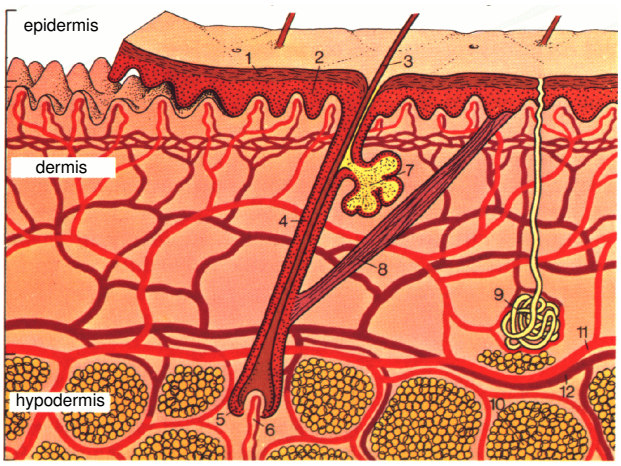


Figure 6.10: **Attachment of muscle fibers to tendons.** Muscle fibers rarely extend over the entire length of the muscle, but insert at an angle into the part of the tendon that runs parallel to the muscle belly.



1	horny layer (*stratum corneum*)
2	germ layer (*stratum germinativum*)
3	hair shaft
4	hair follicle
5	hair bulb
6	accommodative capillary
7	sebaceous gland
8	hair erector muscle
9	sweat gland
10	subcutaneous fatty tissue
11, 12	blood vessels

Figure 6.11: **Cross section of human skin.** Source: [Bau87].

tissue. It contains the sweat glands, hair follicles and blood vessel endings, and provides strong resistance to tearing forces. By constricting or dilating the dermal blood vessels, the body temperature is regulated. The collagen fibers of the dermis are arranged parallel to the skin surface. The wavy bundles uncurl when the skin is stretched. Variance in coiling results in local differences in the resistance to forces. The innermost layer is the *hypodermis*, which is composed of loose connective, mainly adipose (i.e. fatty) tissue. It serves as insulator, storage unit for nutrients, and shock absorber. The skin is connected to the muscle layer by the *fascia*. To some extent, the skin is able to glide over this connective tissue.

Due to the multilayering and the mix of different materials involved, the skin exhibits complicated biomechanical properties [Fun93]. As a basic element of skin, collagen greatly influences its *stress-strain relationship*. This property describes the deformation resistance of a material with respect to the stress applied. The stress-strain relationship of skin has been shown to be non-linear and essentially bi-phasic. Up to a certain point, skin offers low resistance to deformation, because the collagen fibers uncurl and stretch. This behavior changes abruptly once the fibers are stretched fully and aligned to the direction of the applied stress: resistance to deformation increases drastically.
If a constant load is applied to skin tissue, the effects of *creep* can be observed. The skin will continue to elongate, although the applied force does not change. If, on the other hand, the length of the specimen is kept constant, *stress relaxation* causes the internal stress and hence the resistance to stretching of the tissue to decrease over time, i.e. less force is required to keep the

specimen at the desired length. *Hysteresis* is a result of stress relaxation. If stress is first increased and then decreased again, the stress-strain curve of the unloading process will differ noticeably from that of the loading process. If cyclic loading and unloading of skin tissue is performed, the stress-strain curve will change with every repetition. Differences between successive curves decrease and finally disappear. This process is called *preconditioning*.
The above mentioned characteristics illustrate the fact that skin is not truly an elastic solid, but also shares properties with viscous fluids. Such materials are called *visco-elastic*.

6.2 A Physics-based Anatomical Hand Model

This section describes our human hand model with anatomical structure [AHS03], suitable for real time animation using physics-based simulation of muscles and elastic skin properties (Section 6.2.1). The model contains a hybrid muscle model (Section 6.2.2) that comprises pseudo muscles and geometric muscles. Pseudo muscles directly control the rotation of bones based on anatomical data and mechanical laws, while geometric muscles deform the skin tissue using a mass-spring system. Section 6.2.3 proposes a deformation technique based on feature points to warp the complete structure of the reference hand model to an individual hand model derived from a photograph.
Our motivation to choose a physics-based approach controlled through muscle contraction values was the obvious advantage of such an approach: animations are anatomically and physically correct by default. This way, the user need not take care of anatomical or physical limitations but can proceed to design his animations without worrying about the "gory details".

6.2.1 The Reference Hand Model

The central component of our system is a prototype hand model with anatomical structure, which is denoted as the *reference* hand model in the following. The building blocks of our reference hand model are:

- the *skin surface*, which is represented by a triangle mesh consisting of 3000 triangles
- the *skeleton* of the hand, composed of 29 triangle meshes corresponding to the individual bones of the human hand and forearm (cf. Figure 6.12 (right))
- a *joint hierarchy*, which matches the structure of the skeleton, with an individually oriented coordinate system at each joint center defining valid axes of joint rotation
- a set of *virtual muscles*, which are embedded in between the skin surface and the skeleton
- a *mass-spring system*, interlinking the skin, skeleton, and muscles.

Scanning the hand of a living individual is impracticable with current technology, since it is impossible to capture the hand's geometry from all sides while keeping the hand completely immobile. Therefore we used a plaster cast to obtain the skin mesh of our reference hand model, see Figure 6.12 (left). To obtain the mold for the cast, the "model" held her hand into a jar filled with a special mass similar to caoutchouk silicone (Quickform® by Hobby Time®), that within 20 min sets to a rubber-like consistency. Due to the softness of the material, pulling out the hand after this time span was no problem. The mold could immediately be filled with plaster. After the plaster had hardened, the final cast was extracted by peeling away the cast material. Small bumps due to air bubbles in the cast were chipped off with a knife. The cast captures an astonishing amount of detail, even pores are present in the model. This plaster model could

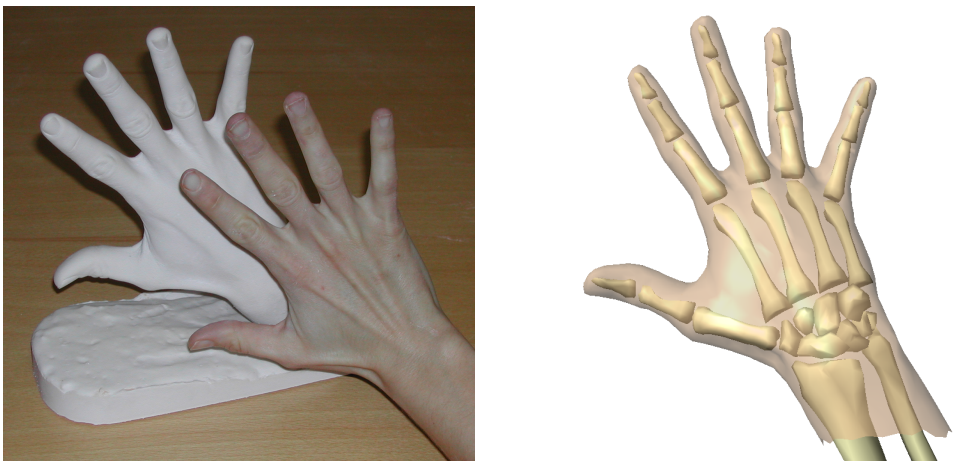


Figure 6.12: **Skin mesh.** Left: the skin mesh of our reference hand model was obtained by scanning a plaster cast of a human hand. Right: the skin of the hand model was rendered semi-transparently to make the underlying bones visible.

be scanned easily. The resulting triangle mesh was reduced to a size of 3000 triangles to allow for real-time simulation of skin deformations. The triangle meshes of the individual bones were taken from a publicly available skeleton model [3D 03] (see Figure 6.12 (right)) and scaled to match the proportions of the skin mesh.
Using the hierarchy of coordinate systems of the individual links, we can model the degrees of freedom for each joint easily. The only joints we ignore are the joints between the individual wristbones. This is justified, since their contribution to the overall movement is negligible. Movement at the intercarpal joint is transferred to the wrist. To overcome the restriction of two orthogonal DOFs, we model the MCP joints of the fingers and the thumb CMC joint as having three DOFs. The muscles must be designed to accommodate the dependencies between the flexion / extension and rotation axes: if a muscle flexes or extends the joint, it must also rotate it to some small degree. The CMC joints of the index and middle finger are fixed, while the ring and little finger CMC joints have two DOFs each with a very small range of motion.
Since muscles usually have greater strength and possible excursion than is required to move the limbs, it is also important to constrain the range of each DOF of the joints to avoid movement which is in reality prohibited by the shape of the joints, by the joint capsules, and by ligaments. For each DOF, we set an upper and a lower limit as listed in Table 6.4.

Animation

Our reference hand model is animated through muscle contraction values given over time. These contraction values are specified as key frames with an arbitrary temporal distribution. During simulation, the contraction values are interpolated using a smooth spline function (Hermite), which is evaluated at discrete points in time according to the desired rendering frame rate. At each point in time, the deformation of all muscles and the position of each bone is computed from the current contraction values. In turn, muscle and skeleton movement is used to update the positions of those nodes of the mass-spring system that attach to muscles and bones, respectively. In the final step of each simulation cycle, the Lagrangian equations of motion are integrated through time for all nodes of the mass-spring system employing a leapfrog Verlet integration method [AT89]. The resulting displacements of the mass nodes attached to the skin mesh effect

joint	DOF	range [°]
wrist	flexion / extension	-50 – 50
	adduction / abduction	-85 – 90
CMC I	flexion / extension	-35 – 55
	adduction / abduction	-40 – 35
	rotation	-10 – 0
MCP I	flexion / extension	-30 – 85
	adduction / abduction	-5 – 5
IP	flexion / extension	-60 – 90
CMC IV	adduction / abduction	-5 – 5
CMC V	adduction / abduction	-5 – 5
MCP II	flexion / extension	-25 – 90
	adduction / abduction	-20 – 30
	rotation	-10 – 15
MCP III	flexion / extension	-25 – 100
	adduction / abduction	-15 – 15
	rotation	-10 – 7
MCP IV	flexion / extension	-25 – 115
	adduction / abduction	-15 – 15
	rotation	-2.5 – 5
MCP V	flexion / extension	-25 – 115
	adduction / abduction	-25 – 15
	rotation	-18 – 5
PIP II-V	flexion / extension	-5 – 110
DIP II-V	flexion / extension	-15 – 90

Table 6.4: **Joint limits.** The listed joint ranges are used in the model. They were measured on the hand that served as model for the template hand. Values are in accordance with [Cha90, LWH00], but may require adjustments for subjects with very different mobility.

the deformation of the skin surface. Details about the geometric muscle model, the mass-spring system, and the integration method can be found in [KHS01], and in Section 3.1.2.

Rendering

Rendering is currently performed using plain OpenGL functionality. Conceptually, it would make no difference to output key frames for a more sophisticated rendering engine. However, we found the possibility to instantly view animations running at real time rates worthwhile enough to accept the somewhat degraded rendering quality. Our focus was on the geometry of the hand model and its deformation during animation, but textures would clearly increase the model's realism.

6.2.2 A Hybrid Muscle Model

Muscle mechanics of the human hand have evolved to a degree of complexity that is unique among mammals. This evolutionary process took place in order to allow us to perform fine motor manipulations and powerful manual work alike. Modeling and simulating all the subtle

anatomical details of the muscles of the human hand is an impractical approach. In this section, we present a *hybrid muscle model*, which is flexible enough to cover the rich variety of muscle mechanics in the human hand and yet runs in real-time.
Our hybrid muscle model comprises *pseudo muscles* and *geometric muscles*. Both of these muscle types are animated exclusively through muscle contraction values within the range $[0, 1]$, where 0 means no contraction at all, and 1 means full contraction. Pseudo muscles directly control the rotation of the bones of the hand, while geometric muscles account for skin tissue deformation through physics-based simulation employing a mass-spring system that connects muscles, skin, and bones. Though each of these two muscle types can be used individually, we typically use a combination of a pseudo muscle and a geometric muscle to represent the effects of an anatomical muscle in the human hand. For instance, the *opponens pollicis* is implemented by a pseudo muscle that rotates the *proximal phalanx* of the thumb and by a geometric muscle (Muscle (2) in Figure 6.15) that bulges the skin. Table A.1 in Appendix A lists the pseudo muscles of our system together with their specific parameters.
For each animation key frame, all pseudo muscles are evaluated to update the position of the bones. The segments of geometric muscles that are attached to bones are transformed correspondingly. Next, the geometric muscles' deformation due to contraction is computed. Finally, the mass-spring system is updated to evaluate the resulting skin deformation.

Pseudo Muscles

Pseudo muscles are virtual muscles that convert a given contraction value $c \in [0, 1]$ into rotation angles φ_k for each DOF of each joint $\mathcal{J}_k$ they affect. Our model for this conversion is based on anatomical data and mechanical laws. However, our implementation is only valid under two assumptions:

1. the bones that are rotated are long bones, which are represented as solid cylinders in our mathematical model. This is true for all bones of the human hand with the exception of the wristbones.
2. when rotating a hierarchy of bones, the number of levels in that hierarchy has to be less or equal to three. In our hand model, this is true for the fingers starting at the knuckles and for the thumb starting at the *trapezium*.

The second restriction is solely due to computational efficiency. Below we describe a technique to efficiently compute the rotation of chains of bones up to length three, which imposes the restriction above. To avoid this limitation, one could apply similarity transforms and the parallel axis (Steiner) theorem [Bar98] for transforming inertia tensors from one coordinate frame to another. This approach removes the limitation of the hierarchy depth at the cost of more expensive computations. In addition, the moment of inertia needs to be stored as a tensor to allow for the application of the parallel-axis theorem for non-parallel rotation axes. In the approach described below, we simplify our model by taking into account only the magnitude of torque and moment of inertia. Treating these variables as vector-valued tensors would render the computational costs of evaluating our model too high for real-time simulation.

Each pseudo muscle represents an anatomical muscle with a given maximum contraction force $\vec{F}_{\max}$. Relative values of $\vec{F}_{\max}$ for all relevant hand muscles are listed in [BBT81]. The direction of $\vec{F}_{\max}$ has to be estimated from the layout and the attachment point of the muscle / tendon (see, e. g., [PP01]). The contraction force of a muscle is not constant, but depends on the current fiber

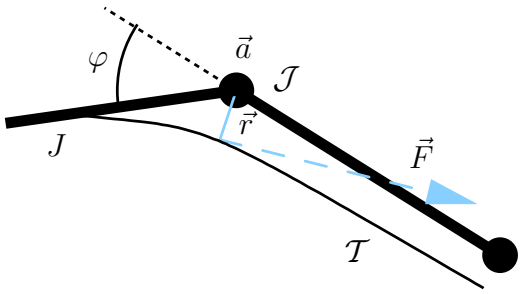


Figure 6.13: **Rotating a single bone.** Tendon $\mathcal{T}$ crosses joint $\mathcal{J}$ at a distance $||\vec{r}||$. The force $\vec{F}$ that $\mathcal{T}$ exerts at $\mathcal{J}$ is tangential to $\mathcal{T}$ and hence perpendicular to $\vec{r}$, but not necessarily perpendicular to $\vec{a}$. $\vec{F}$ causes the distal bone of joint $\mathcal{J}$ to rotate by an angle φ around axis $\vec{a}$, which points out of the image plane. The moment of inertia J of the distal bone depends on this segment's length and mass.

length ℓ of the muscle: a muscle that is either (passively) stretched or (actively) contracted has a lower contraction force than a muscle at its fiber resting length ℓ_0. The non-linear relationship between the contraction force $\vec{F}_{\text{contr}}$ and ℓ is depicted in Figure 6.7 (see also Section 6.1.3). We fitted a quadratic curve to the diagram shown in Figure 6.7 and obtained the relationship:

$$\vec{F}_{\text{contr}}(\ell) \;=\; \left[1 \;-\; 4 \cdot (\ell/\ell_0 - 1.1)^2\right] \cdot \vec{F}_{\max} \;. \tag{6.1}$$

In addition to the contraction force, each anatomical muscle exhibits a stretch force: a muscle that is (passively) stretched counteracts the stretch with a force $\vec{F}_{\text{stretch}}$, which depends on the muscle's current fiber length ℓ. Obviously, the stretch force is equal to zero if $\ell < \ell_0$. Again, we fitted a curve to the diagrams shown in [BH99] and obtained:

$$\vec{F}_{\text{stretch}}(\ell) \;=\; \begin{cases} 2.77 \cdot (\ell/\ell_0 - 1)^2 \cdot \vec{F}_{\max} \;, & \ell \geq \ell_0 \\ 0 \;, & \ell < \ell_0 \;. \end{cases} \tag{6.2}$$

According to [BH99], the inequation $0.6\,\ell_0 \leq \ell \leq 1.6\,\ell_0$ must hold, i.e. a muscle cannot become arbitrarily short or elongated. These upper and lower limits for ℓ are – both in reality and in our model – usually not reached, since the corresponding joints are constrained in their rotations (cf. Section 6.2.1). The current fiber length ℓ of a pseudo muscle is initialized to the resting length ℓ_0 and updated by the arc length of the rotation at the joints it passes.

Rotation of a Single Bone. To see how our conversion model works, let us assume for now that there is exactly one (cylindrical) bone that is rotated about the joint's axis of rotation $\vec{a}$ due to the contraction of one pseudo muscle (Figure 6.13). Given a contraction value $c \in [0, 1]$, the resulting force the muscle exerts on the bone is:

$$\vec{F} \;=\; c \cdot \vec{F}_{\text{contr}}(\ell) \;+\; \vec{F}_{\text{stretch}}(\ell) \;. \tag{6.3}$$

Let $\vec{r}$ denote the moment arm of the force working point. The amount of torque is computed as follows:

$$T \;=\; \text{sgn}(\langle \vec{a}, \vec{r} \times \vec{F} \rangle) \cdot \|\vec{r} \times \vec{F}\| \;, \tag{6.4}$$

where $\langle \vec{a}, \vec{r} \times \vec{F} \rangle$ denotes the dot product of the rotation axis and the vector-valued torque. In addition, the following relationship between torque T, angular velocity ω, and moment of inertia J holds [GV93]:

$$T \;=\; J \cdot \frac{\mathrm{d}\omega}{\mathrm{d}t} \;. \tag{6.5}$$

Since the angular velocity ω equals the first temporal derivative of the rotation angle φ,

$$\omega \;=\; \frac{\mathrm{d}\varphi}{\mathrm{d}t} \;, \tag{6.6}$$

we can discretize time and get:

$$\begin{aligned} \Delta\omega &\overset{(6.5)}{=} \Delta t \cdot J^{-1} \cdot T \;, \\ \omega_{\mathrm{new}} &= \omega_{\mathrm{old}} + \Delta\omega \;, \\ \Delta\varphi &\overset{(6.6)}{=} \Delta t \cdot \omega_{\mathrm{new}} \;, \\ \ell &\leftarrow \ell - \Delta\varphi \cdot \|\vec{r}\| \;. \end{aligned} \tag{6.7}$$

Using this approach, we can compute the increment $\Delta\varphi$ of the rotation angle from a contraction value c. The only unknown variable is the moment of inertia J. Although J is quite expensive to compute for an arbitrarily shaped body, it can be easily computed for a solid cylinder of length l and mass m that is rotated about an axis orthogonal to its length axis and passing through one of its ends [Gol02]:

$$J \;=\; \frac{1}{3} \cdot m \cdot l^2 \;. \tag{6.8}$$

In our case, the length l is the length of the bone rotated about the joint's axis. The mass m is the mass of the bone plus the mass of the tissue surrounding the bone. Values for this bone-plus-tissue mass can be found in [BY94].

The above formulas do not consider friction yet. This means that a rotation, once it has started due to muscle contraction, will not stop again. To take into account friction, we have to modify Equation (6.4) by subtracting the torque of friction:

$$T \;=\; \mathrm{sgn}(\langle \vec{a}, \vec{r} \times \vec{F} \rangle) \cdot \Big[\|\vec{r} \times \vec{F}\| \;-\; \mu \cdot |\omega_{\mathrm{old}}| \Big] \;, \tag{6.4$'$}$$

where μ is the coefficient of friction. In accordance to medical literature, we use $\mu = 0.015$.

Finally, we extend our mathematical model to allow for an arbitrary number n of pseudo muscles that affect the rotation of the bone. Each pseudo muscle i $(i = 1, \ldots, n)$ exerts the force:

$$\vec{F}_i \;=\; c_i \cdot \vec{F}_{\mathrm{contr},i}(\ell_i) \;+\; \vec{F}_{\mathrm{stretch},i}(\ell_i) \;. \tag{6.3$'$}$$

Since each muscle has its own moment arm $\vec{r_i}$, the total amount of torque is given by:

$$T \;=\; \mathrm{sgn}(\langle \vec{a}\,, \vec{T} \rangle) \cdot \Big[\|\,\vec{T}\,\| \;-\; \mu \cdot |\omega_{\mathrm{old}}| \Big] \;, \tag{6.4$''$}$$

with

$$\vec{T} \;=\; \sum_{i=1}^{n} \vec{r_i} \times \vec{F_i} \;. \tag{6.9}$$

Values for fiber resting length, affected DOFs, and moment arm of the individual muscles are assembled in Table A.1 in Appendix A.

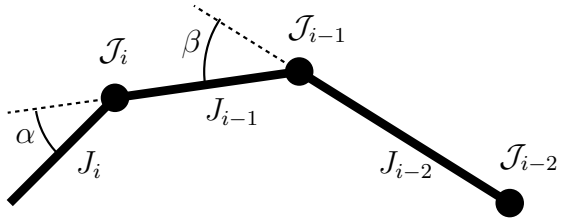


Figure 6.14: **Moments of inertia for a chain of bones.** $\mathcal{J}_{i-2}$, $\mathcal{J}_{i-1}$, and $\mathcal{J}_i$ are consecutive joints with moments of inertia J_{i-2}, J_{i-1}, and J_i, respectively. α denotes the rotation angle of joint $\mathcal{J}_i$, and β is the rotation angle of joint $\mathcal{J}_{i-1}$.

Rotation of Chains of Bones. For the derivation of the conversion formulas in the previous paragraph we assumed that the rotated bone is an end segment, i.e. one of the *distal phalanges*. If, however, we want to rotate a chain of bones, for instance the three *phalanges* of a finger, the moment of inertia J depends on the position of all bones in that chain.
Figure 6.14 depicts this situation: when rotating about joint $\mathcal{J}_i$, the moment of inertia J_i of the rotated bone is constant and can be computed according to Equation (6.8). The total moment of inertia for a rotation about joint $\mathcal{J}_{i-1}$ is composed of J_{i-1} and the moment of inertia of the end segment. The latter, however, is not simply J_i in this case: the axis of rotation does not pass through the end of the rotated end segment as required for Equation (6.8). Thus the position of the end segment has to be transformed into the coordinate system of $\mathcal{J}_{i-1}$ and the moment of inertia J_i^* is computed by summing up the squared distances of the transformed bone mesh vertices to the rotation axis multiplied by the mass of the bone. This computation becomes more and more costly when longer chains of bones are rotated.
Fortunately, the moment of inertia J_i^* of the transformed bone depends only on the rotation angle α. Thus we precompute $J_i^*(\alpha)$ for a discrete set of angles (typically in steps of five degrees) and store the array $J_i^*[\alpha]$ in the joint $\mathcal{J}_i$ for further look-up. The total moment of inertia J for a rotation about $\mathcal{J}_{i-1}$ can thus be simply computed as $J = J_{i-1} + J_i^*[\alpha]$. Similarly, the total moment of inertia for a rotation about $\mathcal{J}_{i-2}$ is given by the sum $J_{i-2} + J_{i-1}^*[\beta] + J_i^*[\alpha][\beta]$.
Precomputing the moments of inertia for chains of bones with more than three segments would require storing arrays of dimension three and more in the joints. To avoid this exhaustive memory consumption, we restrict the computation of the moments of inertia to hierarchies with at most three levels. For the rotation of the complete hand about the wrist we assume a constant moment of inertia of the hand.

Geometric Muscles

In our system, geometric muscles are embedded in between the skin surface and the underlying bone structure. Geometric muscles have an actual geometric shape assigned to them, which deforms and bulges during contraction. Springs are used to connect the surface of the muscle to skin and bones. We have adopted the approach presented in [KHS01] (see Section 3.1.2) for the embedding of muscles into a mass-spring system. However, some modifications of that approach were necessary to allow for a more complex muscle layout. In particular, we have introduced the following changes:

– each individual muscle has its own minimum and maximum thickness. Rather thick muscles, e.g. the *opponens pollicis* (2), can thus be created as well as thin sheet muscles, e.g.

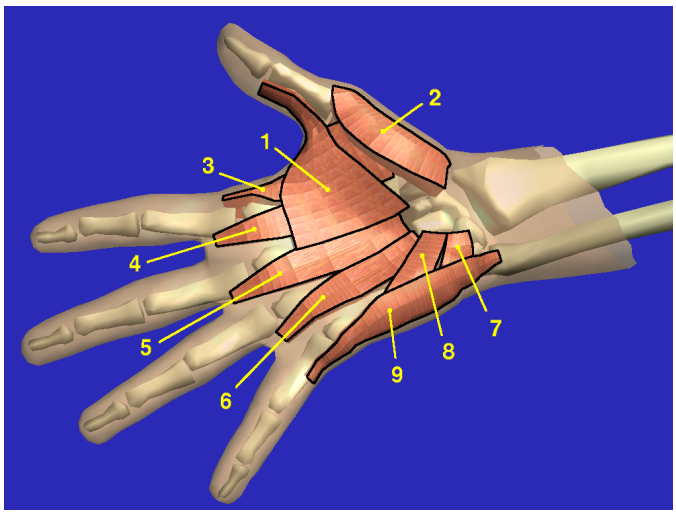


Figure 6.15: **Geometric muscles of our hand model.** *adductor pollicis* (1), *opponens pollicis* (2), *1st dorsal interosseus* (3), *1st palmar interosseus* (4), *2nd palmar interosseus* (5), *3rd palmar interosseus* (6), *opponens digiti minimi* (7), *flexor digiti minimi brevis* (8), and *abductor digiti minimi* (9).

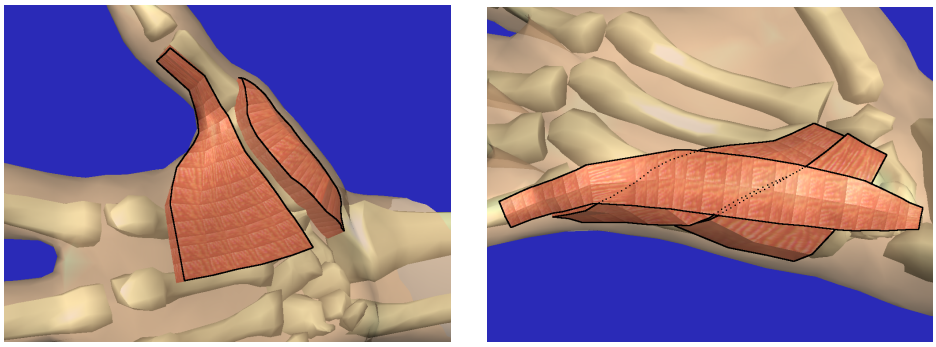


Figure 6.16: **Muscle Layout.** Left: the complex shape of muscles can be observed in this close-up view of *adductor pollicis* and *opponens pollicis* with all other muscles removed. Right: different muscle layers are set up automatically. The vertical muscles *opponens digiti minimi* and *flexor digiti minimi brevis* slide freely below the horizontal *abductor digiti minimi*.

the *adductor pollicis* (1), see Figure 6.15.

- the distance between the skin surface and the surface of the muscle can be set individually for each muscle to allow for several layers of muscles (e.g. superficial and deep layer) to be created automatically, see Figure 6.16 (right).
- muscles are allowed to attach to bones on both muscle ends. Such types of muscles do not exist among the facial muscles (with the exception of the *masseter*, which was not present in [KHS01]), but are prevalent in the human hand.
- muscles may be assigned to several individual bones. Thus, individual segments of large or long muscles move with the bones they are assigned to. The *abductor digiti minimi*, for instance, is assigned to the *carpal bones* (wristbones), and the *metacarpal* and *proximal phalanx* of the little finger (cf. Figure 6.16 (right)).

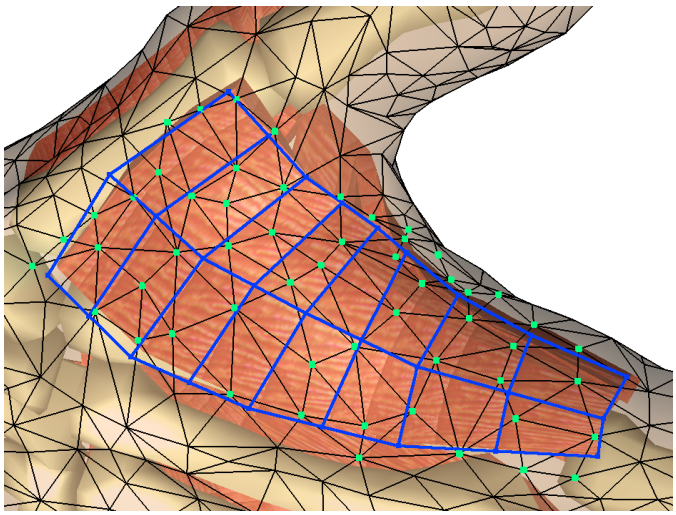


Figure 6.17: **Muscle grid.** Geometric muscles are created automatically from a muscle grid (shown in blue) painted onto the skin surface. The green dots mark the vertices of the skin mesh (shown as a wireframe), which are influenced by the muscle.

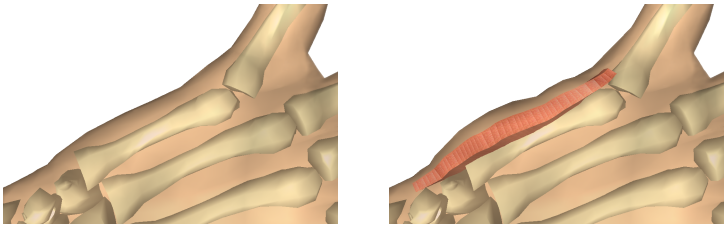


Figure 6.18: **Bulging of geometric muscles.** Left: pseudo muscles are used to move the bones. Right: combining pseudo and geometric muscles results in additional skin deformation due to bulging.

Geometric muscles are created by interactively painting *muscle grids* onto the skin surface (see Figure 6.17). From the shape of a muscle grid, the corresponding muscle is created automatically to fit in between skin and bone surfaces. This fitting process is analog to the one described in [KHS01], see p. 19 in Section 3.1.2, but uses for every muscle individual parameters for muscle thickness and skin distance. The attachment process of the muscle control points to the skeleton works in the same way as for the MEDUSA models, except that there are more than two candidate bones.
Modeling geometric muscles that are truly attached to bone on both ends would mean that contracting a muscle moved the bone into which it inserts. This would make the mass spring network considerably more complicated, which was the reason why we chose to give every geometric muscle a "loose" end like the facial muscles in MEDUSA that insert into skin. Instead we introduced the pseudo muscles. The reference hand model has a pseudo muscle for every geometric muscle. Conversely, providing a geometric muscle for every pseudo muscle is not necessary, since most muscle bellies are located in the forearm. Passive deformation of the geometric muscles due to bone rotation is ensured by the attachment of the muscle control points to the underlying bones. The same transformation is applied to the muscle grid points as to the bone, thereby deforming the respective muscle segments.

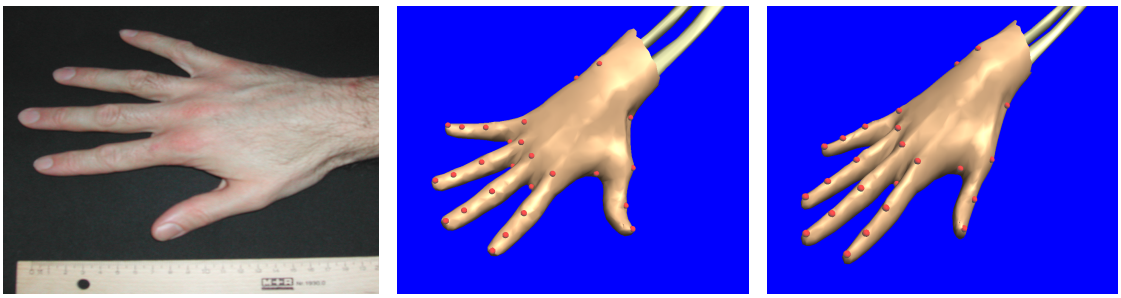


Figure 6.19: **Deformation of the reference hand model.** Left: photograph of an individual's hand, including a ruler. Middle: position of the 26 source feature points on the reference hand model. Right: resulting hand model after applying the warping function.

During contraction of a geometric muscle, the assigned geometric shape deforms and bulges, see Figure 6.18. On the muscle surface, the attachment points of the springs connecting the muscle to the skin move accordingly. The displacement of these nodes exerts force on the mass-spring mesh, which is updated to compute the corresponding skin deformation (cf. Section 6.2.1).

6.2.3 New Hand Models from Photographs

We employ a deformation technique based on feature points to warp the complete reference hand model to an individual hand model. This saves the user the trouble of rigging new models by hand. Our approach is similar in spirit to the technique proposed by Kähler et al. [KHYS02] (see also Section 3.1.2, p. 24) for deformation of human head models. However, we do not require a 3D target hand model to be obtained in a time-consuming scanning process. Instead, we use a photograph of the individual hand to be modeled. The photograph merely needs to show a simple ruler as depicted in Figure 6.19 (left). Since there are no other prerequisites for the photograph, low-cost consumer cameras can be used for the acquisition.
First, we identify a small set of feature points in the input photograph. Our reference hand model is already tagged with the same feature points by default. Next, the complete structure of the reference hand model is deformed to match the shape of the individual hand from the photograph. The warp function is set up using correspondence of feature points. Below, each step is described in more detail.

Feature Points

We use a small set of feature points on both the 3D reference hand model and in the 2D photograph. These feature points can be easily identified without anatomical knowledge. In the following, the feature points on the reference hand model are denoted as the *source feature points*, whereas those in the input photograph are called *target feature points*.
The reference hand model is equipped with 26 source feature points by default (cf. Figure 6.19 center), as listed in Table 6.5.
Upon loading the input photograph into our system, the user is asked to identify as many target feature points from the above set as possible in the photograph. Feature points whose positions are not clearly visible in the photograph can be omitted. In our experiments, we obtained reasonable results using a subset of only 16 feature points.

#	location
5	finger tip I-V
4	DIP II-V
1	IP I
4	PIP II-V
5	MCP I-V
4	interdigital skin (between each pair of adjacent fingers)
2	radial and ulnar wrist (inner and outer side of the wrist)
1	head of ulna

Table 6.5: **Feature points.** The reference hand model comes equipped with these 26 feature points.

In addition to selecting the target feature points, a calibration process is carried out to measure the size of the hand in the input photograph. To this end, the user performs two mouse clicks at a known distance on the division scale of the ruler shown in the photograph, for instance at the points "0 cm" and "20 cm". From their pixel distance d on the photograph, the system automatically computes the scale of the photograph $s = \frac{20\,\text{cm}}{d\,\text{pixels}}$ and uses this information to transform the positions of the target feature points into the coordinate system of the reference hand model.

Warping the Reference Hand Model

Given two sets of N corresponding source and target feature points, we are looking for a function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$ that maps the source feature points $\mathbf{s}_i$ to the target feature points $\mathbf{t}_i$, i.e.

$$f(\mathbf{s}_i) = \mathbf{t}_i, \quad i = 1, \ldots, N \; .$$

A natural solution to this interpolation problem is to employ a radial basis function (RBF), see for instance [CBC+01] for mathematical details. A RBF consists of N basis functions φ_i, defined by the source feature points $\mathbf{s}_i$. Hence

$$f(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{c}_i \varphi_i(\mathbf{x})$$

with (unknown) weights $\mathbf{c}_i \in \mathbb{R}^3$. We use biharmonic basis functions $\varphi_i(\mathbf{x}) := \|\mathbf{x} - \mathbf{s}_i\|_2$, which minimize bending energy [Duc77]. This choice is in consonance with Bookstein's suggestion to use thin-plate splines for the deformation of biological tissues [Boo97a, Boo97b].

Before setting up the function f, we need to transform the target feature points $\mathbf{t}_i$ into the coordinate system of the $\mathbf{s}_i$. This is necessary to make the transformed hand model appear in approximately the same place as the initial reference hand model. We transform the 2D points $\mathbf{t}_i$ into the fitting plane of the 3D points $\mathbf{s}_i$ using a rigid body transformation. The (uniform) scaling factor of this transformation is taken from the calibration step described above.
To compute the fitting plane, let

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i$$

be the center of gravity of the source feature points. Then we shift the $\mathbf{s}_i$ so that their center of gravity coincides with the coordinate system's origin. Now define $\mathbf{X}$ to be the matrix with the shifted $\mathbf{s}_i$ as columns:

$$\mathbf{X} = (\mathbf{s}_i - \mathbf{c})_{i=1}^{N} \in \mathbb{R}^{3\times N} .$$

The eigenvectors of the two largest eigenvalues of the covariance matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}$ span the fitting plane, while the third eigenvector is the normal $\mathbf{n}$ of the plane. Since $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ is symmetric and hence diagonalizable, the three eigenvectors differ pairwise.

The normal $\mathbf{n}$ and the center of gravity $\mathbf{c}$ of the $\mathbf{s}_i$ describe the fitting plane. Now the target coordinate system with the $\mathbf{t}_i$ in the xy-plane is translated to $\mathbf{c}$ and its z-axis is rotated so that it becomes parallel to $\mathbf{n}$. Lastly, the target feature points are rotated to align the largest diameter of their set (between the feature point on the tip of the index finger and the feature point on the head of the *ulna*) to the largest diameter of the set $\{\mathbf{s}_i \,|\, i = 1\ldots, N\}$.
Next, the target feature points are lifted to the third dimension by assigning to each $\mathbf{t}_i$ the z coordinate of its corresponding source feature point. Remember that the z coordinates define the distance of the feature points from the fitting plane. These heights are additionally scaled by a (uniform) scaling factor obtained from the ratio of the largest diameters of the source and target feature points, respectively. Converting the 2D target feature points into 3D points is essential to ensure that the transformed hand model will not be flattened but possesses a thickness proportional to its overall size. Finally, we can set up our radial basis warping function as described in standard literature [PHL+98, CBC+01]. To obtain the $\mathbf{c}_i$, let

$$\begin{aligned}
\mathbf{\Phi} &= \begin{pmatrix} \varphi_1(\mathbf{s}_1) & \ldots & \varphi_N(\mathbf{s}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{s}_N) & \ldots & \varphi_N(\mathbf{s}_N) \end{pmatrix} && \in \mathbb{R}^{N\times N} \\
\mathbf{C} &= (\mathbf{c}_1, \ldots, \mathbf{c}_N)^T && \in \mathbb{R}^{N\times 3} \\
\mathbf{T} &= (\mathbf{t}_1, \ldots, \mathbf{t}_N) && \in \mathbb{R}^{3\times N} .
\end{aligned}$$

Now we can set up a system of linear equations $\mathbf{\Phi C} = \mathbf{T}$ and solve for the $\mathbf{c}_i$ using, for example, standard LU decomposition with pivoting, and set up the warping function f.

RBFs are defined on the volume spanned by the feature points and can hence be used to deform the complete structure of our reference hand model. Hereby we proceed as follows:

1. skin and bone meshes are transformed by applying the function f to each vertex of the meshes. The connectivity of the meshes is not changed.
2. joint positions and positions of feature points are transformed in the same way by direct application of the warping function.
3. the coordinate axes of the joint frames must be handled differently: the new z-axis of joint $\mathcal{J}_i$ is the unit vector pointing from the new position (origin) of $\mathcal{J}_i$ towards the transformed origin of its child $\mathcal{J}_{i+1}$. The corresponding transformation that rotates the old z-axis onto the new z-axis is applied to all three axes. Finally, the frame is rotated around the z-axis so that the x- and y-axis are as closely aligned to their old counterparts as possible. A final manual adjustment may be necessary if the shape of the new hand differs very much from the reference hand.

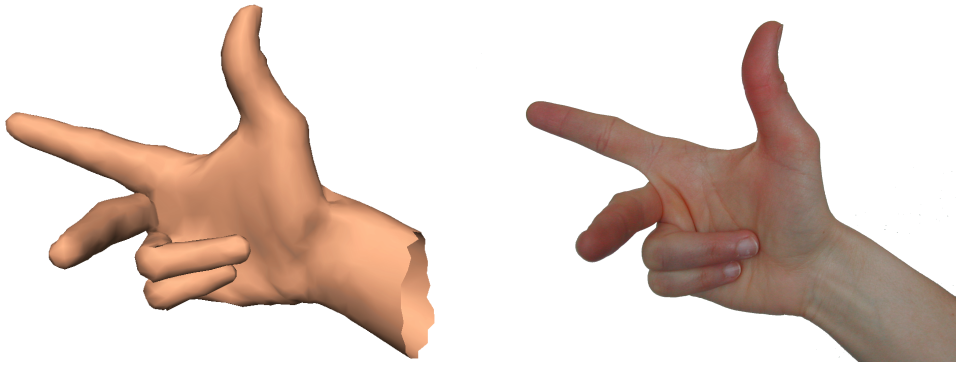


Figure 6.20: **A typical computer graphics scientist's hand pose.** Side-by-side comparison of our hand model (left) with a photograph of the hand that was scanned to build the model (right).

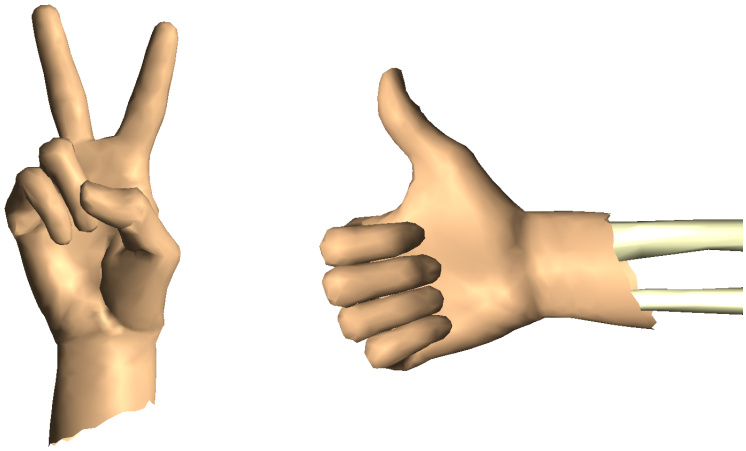


Figure 6.21: **Optimists only.** These hand poses are animation snapshots from our system.

4. geometric muscles are defined by muscle grids (cf. Section 6.2.2). To warp the muscles, only the control points of the muscle grids have to be transformed. The shape of the warped muscles is computed automatically to fit in between the transformed skin and bone meshes.
5. pseudo muscles do not have any spatial parameters that need to be transformed.

6.2.4 Results

We created several animation sequences to verify the efficiency and correctness of our hybrid muscle model. In spite of some simplifications, we found our pseudo muscle model to work well and yield plausible results. Figure 6.20 shows a side-by-side comparison of our hand model with a photograph of the hand that had been scanned to build the reference hand model. Clearly, our hybrid muscle model is able to correctly reproduce the pose of the hand. Additional animation snapshots are depicted in Figure 6.21.
When creating a new animation, however, estimating the right muscle contraction values (i.e. those that result in the desired movement of the fingers) is not always a straightforward process. Employing an optimization process that computes minimal energy muscle contractions for

a given target position of the fingers would eliminate the process of specifying individual muscle contraction values. Creating the geometric muscles is simple and fast: the complete set of geometric muscles shown in Figure 6.15 has been created in less than an hour.
Using our hybrid muscle model, animations were running at real time frame rates of over 60 fps on a 2.4 GHz AMD Opteron 250 dual processor machine with a NVidia Quadro NVS graphics board. The main bottleneck is the integration of the equations of motion for the mass-spring network, which comprises approximately 1500 nodes and 4500 springs. Yet our integration technique runs stable for a reasonable choice of stiffness parameters for the skin model. In all our tests, the skin mesh never lost integrity after mass-spring simulation. In particular, our method does not break the skin while moving from pose to pose, since the connectivity of the skin mesh is never changed.
Our deformation technique works reliably and is easy to use. We warped our reference hand model, built from scan data of a female, to match the size and proportions of a man's and a child's hand. The interactive specification of the target feature points in the photograph takes about 1 min. Since all components of the hand model are transformed, the resulting hand model is instantly animatable. However, the animation parameters (i.e. the muscle contraction values) have to be adapted to the warped hand model: different proportions of reference and target hand model result in different torques, moments of inertia, and consequently different rotation angles.

6.2.5 Conclusions

This section dealt with an approach for the construction and animation of human hand models with underlying anatomical structure. Our system is built around a reference hand model, which is animated using muscle contraction values. We introduced a hybrid muscle model that comprises pseudo muscles and geometric muscles. While pseudo muscles control the rotation of bones based on anatomical data and mechanical laws, the deformation of geometric muscles causes realistic bulging of the skin tissue. As a result, the created animations automatically exhibit anatomically and physically correct behavior. In addition, we proposed a deformation technique based on feature points to create individual hand models from photographs. Warping the complete structure of the reference hand model results in deformed hand models that are instantly animatable.

Although our system is working reliably and rather efficiently, there are further ways of improvement. Ideally, geometric muscles should move the bones. This, however, involves modeling tendons as well as setting up a mass-spring system where rigid objects (such as the bones) can be moved due to spring forces. In addition, gravity should be included into our hybrid muscle model. While the effect of gravity is probably negligible for the deformation of skin tissue in the human hand, it plays an important role for the computation of bone positions from given muscle contraction values.
To facilitate the creation of animation sequences, optimization could be used to compute minimal energy muscle contractions for a given target position of the fingers. Moreover, it would be helpful to include collision detection among the parts of the hand. Given these two add-ons (optimization process and collision detection), grasping of external objects would be rather easy to implement.
Finally, it would be desirable to automatically generate textures from the same photographs that are used to create individual hand models. However, it is not trivial to compute the parameterization of the skin mesh fully automatically, if the texture is to be used for OpenGL rendering.

6.3 An Application: the Pitcher's Hand

The hand model from Section 6.2 was used to visualize the hand poses measured by a high-speed tracking system based on multi-exposure photography [TAH+05]. We developed this system in order to address the problem of capturing and tracking high-speed motion sequences that cover large areas of space. To avoid the expenses of professional high-speed video cameras and high-resolution motion capture equipment, our approach is based on low-cost commodity still cameras and strobe lights.
As an example application, we configured the system to track baseball pitches. Due to its variety of different elements, baseball is technically very challenging. In particular, pitching is the single most important part of baseball. The goal for the pitcher is to throw the ball in such a way that its trajectory is as unpredictable as possible for the other team's batter. In the history of baseball a great variety of pitches has been developed. The art of pitching is to be able to perform all kinds of pitches such that the ball consistently enters the strike zone near the batter in order to be valid. This sport offers itself as a test case for our motion capture system because it is ideal to demonstrate the strengths of the system. First of all, the underlying motion is very fast and extends over a large area of space: the speed of a pitched baseball can reach 120 km/h and above, and the distance from the pitcher mound to the home base is 60.5 feet (18.44 meters). In addition, there are many different motion parameters that can be measured simultaneously for a variety of pitches:

- pose parameters of the pitcher's hand before, at, and after releasing the ball
- 3D trajectory of the flying ball
- initial flight parameters of the ball: norm and direction of initial velocity, rotation axis, spin.

With a physically based model of ball flight, we were able to demonstrate the accuracy of the system. Since even a small difference in the pose of the pitching hand can have a big impact on the ball trajectory, exactness of acquisition and visualization is of utmost importance.

In our experiments we focused on the following pitches: the fast-ball, the curveball, the slider, and the change-up, all performed as three-quarter deliveries, i.e. with a release point above and to the right of the head. Each of these pitches was recorded multiple times. Pitches differ in the way how the hand moves during launch, giving the ball a different initial velocity, rotation axis, and spin. Since these initial flight parameters completely determine the ball's trajectory, different pitches lead to different flight paths.

- the fast-ball is the fastest pitch. It has large back spin and, depending on whether the ball rotates over four or only two of its seams, it is called a 4-seamer or a 2-seamer.
- the change-up also exhibits back spin but has a lower velocity and spin.
- the curveball is released with forward spin which makes the ball descend faster during the last phase of its flight.
- the slider is thrown with a side spin, making the ball turn to one side towards the end of the flight.

All pitches were carried out by a professional baseball pitcher who is able to perform different pitches with great faithfulness.

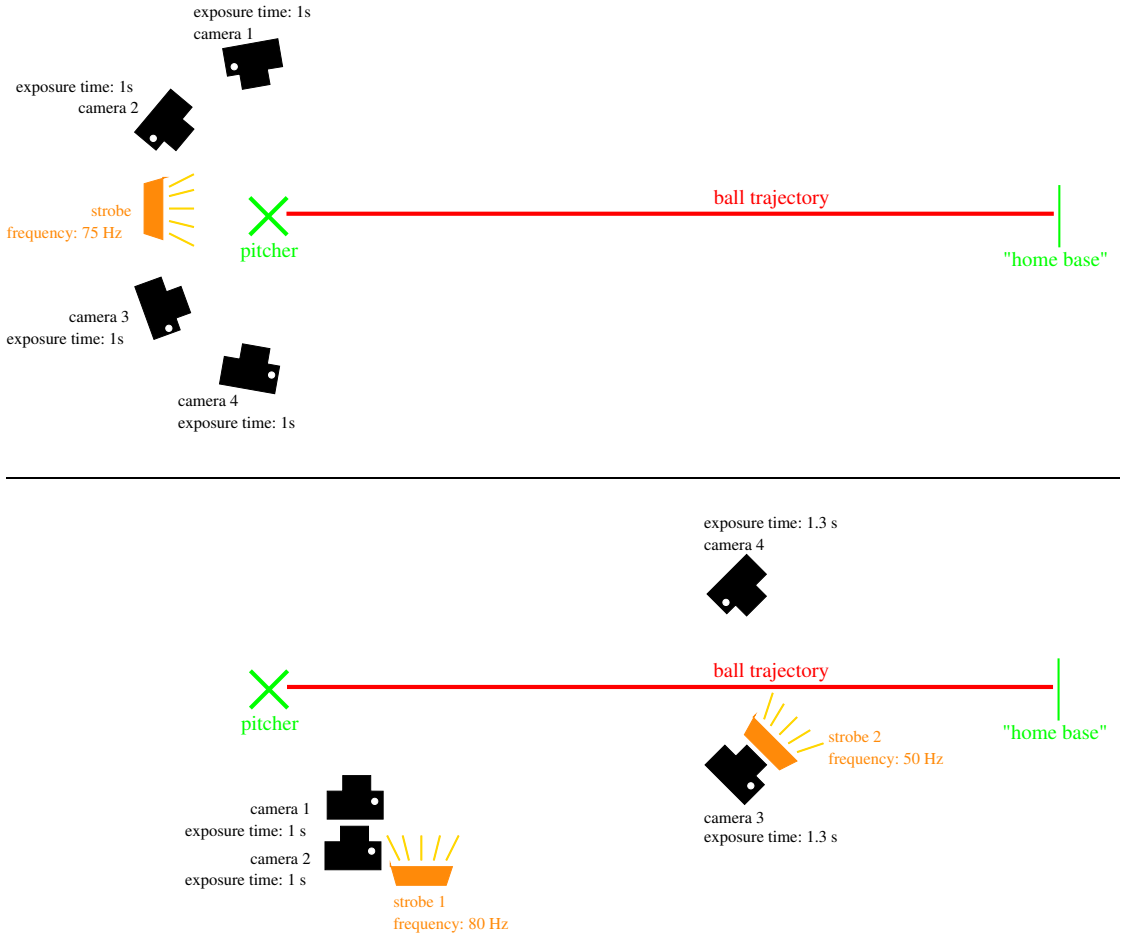


Figure 6.22: **Setup.** Top: setup for tracking the ball and the hand at ball release. Bottom: setup for capturing initial flight parameters (cameras 1&2, strobe 1) and ball trajectory (cameras 3&4, strobe 2). See Figures 6.23 and 6.24 for photographs of the setup.

In this section, we will first give an overview of the tracking system (Section 6.3.1). This is succeeded by a description of the methods used for capturing and visualizing hand (Section 6.3.2) and ball motion (Section 6.3.3). Our results are presented in Section 6.3.4, and conclusions are drawn in Section 6.3.5.

6.3.1 System Overview

A flexible setup permits us to robustly acquire different types of motion data under real-world conditions. We want to capture the motion of the pitcher's hand and fingers before, during, and after releasing the ball. This is achieved by tracking markers on the highly mobile pitching hand. A setup that minimizes occlusion of the hand is important here. To analyze flight trajectories of different pitches we need to acquire image data that allows us to reconstruct the ball's initial flight parameters (i.e. norm and direction of its velocity, direction of its rotation axis, and spin) as well as the 3D positions of the ball along its trajectory. Acquiring this type of information is very challenging since the involved speeds are considerable and the entire trajectory extends over a relatively wide area. To complicate things even further, high spatial accuracy is essential

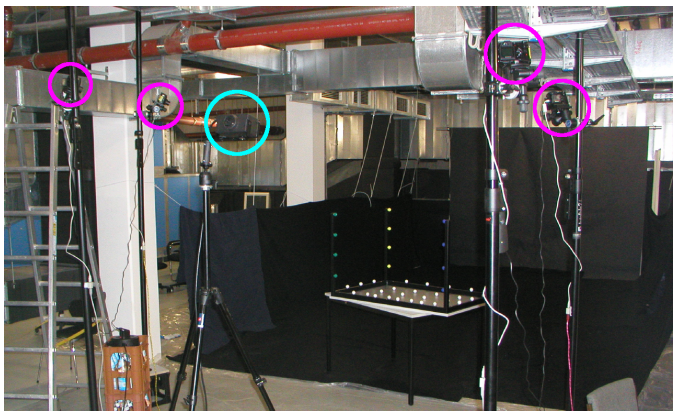


Figure 6.23: **Setup for hand pose acquisition.** Two stereo camera pairs (magenta) and a strobe light (cyan) are placed in a semi-circular arrangement around the pitcher to capture the hand motion of different pitches. In the center, the calibration object used to estimate the extrinsic camera parameters can be seen. See Figure 6.22 (top) for a schematic illustration of the setup.

in both hand motion capture and ball flight analysis.
To capture an entire baseball pitch, we set up our acquisition gear in a basement room which has a central free space of approximately 25 m length, 4 m height, and 5 m width. This is sufficient to house the complete pitching corridor (18.44 m in length) as well as to put up the camera and lighting equipment. As imaging devices we employ consumer-market Olympus™Camedia C5050 still image cameras that provide a frame resolution of 2560x1920 pixels. This camera model features a large-aperture zoom lens that can be set to a comparatively wide angle. We use four cameras of this type in our setup. The settings of all four cameras are controlled from a single PC, which triggers all camera shutters simultaneously.
Since we intend to record a fairly wide-area scene, we need a sufficiently luminous stroboscope light source that can illuminate a large volume at high frequencies. In our setup we use two high-output strobe flashes which have an intensity of 5000 Lux each at a distance of 0.5 m from the lamp. At full intensity, the 20 μs-long flashes can be triggered at up to 80 Hz, which is sufficiently fast for our purposes.
During recording the floor and walls are covered with black carpet and cloth to facilitate foreground object segmentation and automatic marker tracking. Primarily, however, the dark material absorbs most light that has not hit foreground objects, preserving contrast and preventing quick saturation of the multi-exposure images. A heavy dark carpet hanging down from the ceiling at the end of the flight corridor absorbs the impact of the ball.

In our recordings, four simultaneously triggered cameras look at the scene from different positions. Two different arrangements of imaging sensors and light sources are needed to record either the hand motion of the pitcher or the initial flight parameters and ball positions.
For recording the hand, the four cameras and one light source are placed in a semi-circular arrangement looking at the pitcher from behind and above, see Figures 6.22 (top) and 6.23. Section 6.3.2 gives further details about this step.
To record the baseball in flight, two stereo pairs of cameras and two stroboscopes are used to capture the initial and final phase of the ball flight, respectively (Figures 6.22 (bottom) and 6.24). Details about the setup for acquisition of ball motion are given in Section 6.3.3.

Figure 6.24: **Ball acquisition setup.** Left: a stereo camera pair (encircled in magenta) facing the black curtain on the right is capturing the ball's initial flight parameters. The ball is illuminated by a stroboscope (cyan). Right: a stereo camera pair (magenta) and a strobe light (cyan) facing towards the black carpet in the back are responsible for capturing the ball trajectory close to the "home base". See Figure 6.22 (bottom) for a schematic illustration of the setup.

Accurate calibration of the cameras is crucial. We apply a camera model for short focal length cameras [HS96]. Intrinsic camera parameters are estimated from images of a planar checkerboard pattern. Radial and tangential lens distortion are modeled up to second order [JKS95]. Each multi-exposure image is distortion-corrected prior to any further processing. Extrinsic camera parameters are estimated using images of our 3D calibration object, see Figure 6.23. Camera position and orientation are metrically calibrated.

Finally, we rely on our professional baseball pitcher who, as we have verified, performs different pitches with great faithfulness. This allows us to correlate our measurements of hand motion with the measurements of initial flight parameters and flight trajectory.

6.3.2 Tracking the Hand

This section details how to reconstruct the pitcher's hand poses by means of the multi-exposure technique. For visualization purposes, animations from the acquired poses were generated with the physics-based hand model described in Section 6.2.

Preparation of the Pitcher's Hand

In order to determine the locations of the finger joints in the recorded images, we have to mark them on the pitcher's hand. Therefore the pitcher wears a thin, transparent rubber glove onto which colored markers made of reflective tape are glued, see Figure 6.25 (left). The markers are placed on the joint positions, on the finger nails, and on three distinct positions on the back of the hand. Four different marker colors are distributed such that the distance between any two markers of the same color is maximized. In total, 18 positions on the hand are tagged and assigned a unique position label. To facilitate identification of the markers in the multi-exposure images, the skin underneath the glove is painted with black make-up. During recordings, the pitcher wears black clothes and a black face mask to prevent misclassifications.

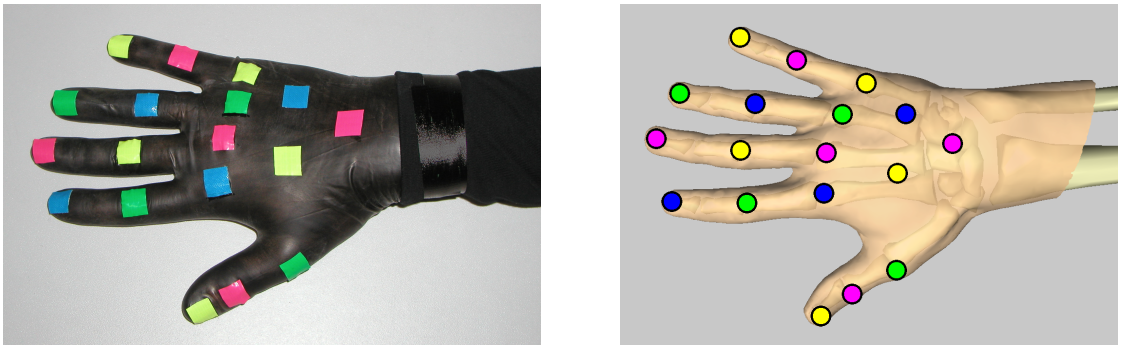


Figure 6.25: **Hand markers.** Left: markers for tracking are attached to the pitcher's hand. Right: corresponding marker positions on the personalized hand model.

Our pitcher assured us that the glove did not impair his grip on the ball or his freedom of movement. Thus, pitches where he was wearing the rubber glove did not differ from those without.

Acquisition of Hand Motion

For acquisition of hand motion, all four cameras and one stroboscope are positioned in a semi-circular arrangement behind the pitcher, see Figure 6.22 (top). In front of the pitcher, the walls and the floor of the flight corridor are covered with black cloth. All cameras are focused on the region where the pitcher releases the ball. The camera positions are chosen in such a way that two cameras observe the hand motion from the pitcher's left and two from the pitcher's right side. This way occlusions of the hand markers during the complex pitching movement are minimized and sufficiently separated exposures of the hand in the images are obtained. The strobe light is located directly behind and slightly above the pitcher such that the focus of illumination coincides with the release position of the ball. During recordings the stroboscope operates at 75 Hz, a frequency that leads to a high number of visible hand positions sufficiently separated in the images for all pitch types. For recording, all four cameras are triggered synchronously with an exposure time of 1 s. As a trade-off between image noise and brightness, we run each camera with ISO 200 sensitivity. We have recorded the four types of pitches described above.

Tracking Hand Positions

First we separate the marker positions from the background in each of the four multi-exposure images using background subtraction. Since all unimportant parts of the scene are colored black, the reflective markers emerge very brightly in the images, see Figure 6.26. In order to identify the locations of the markers in each photograph, a color interval for each marker type is defined. Connected image regions above a minimum size whose pixels fall into one of the intervals are considered as projected marker locations[1]. The projected centers of the markers are approximated as the centers of gravity of the marker regions. Correspondences between different camera views are established via epipolar lines [Fau93]: for stereo camera pairs, each image point in one camera view has a corresponding point in the other camera view which lies somewhere along the epipolar line of the second image. This concept is illustrated and explained in Figure 6.27. Tech-

[1]The marker locations in the photograph are projections of the real world marker positions onto the camera's image plane.

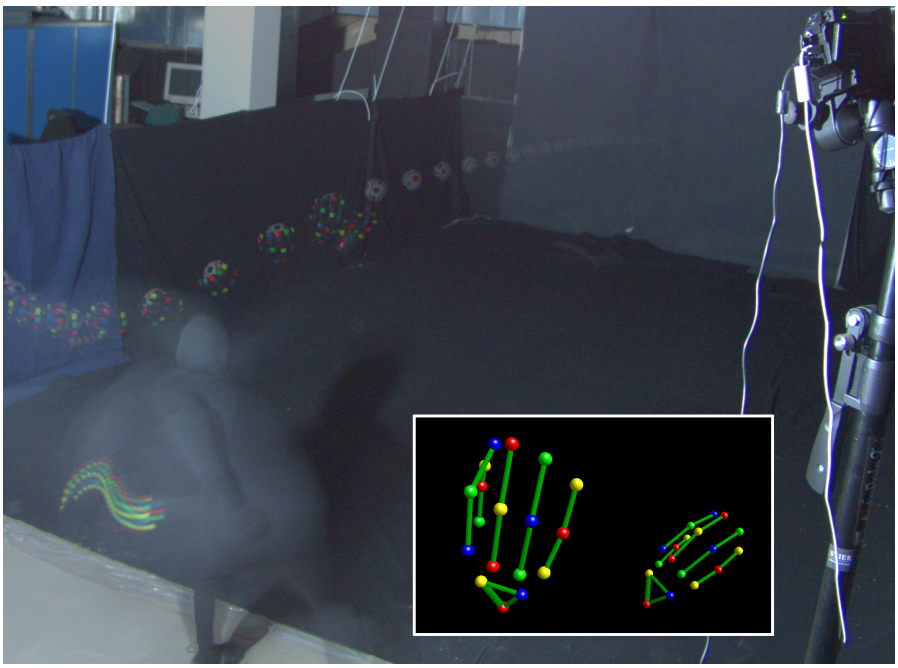


Figure 6.26: **Multi-exposure image of the hand.** This multi-exposure image from one of our cameras records the hand motion during pitching. Inset: reconstructed hand marker positions for two hand poses.

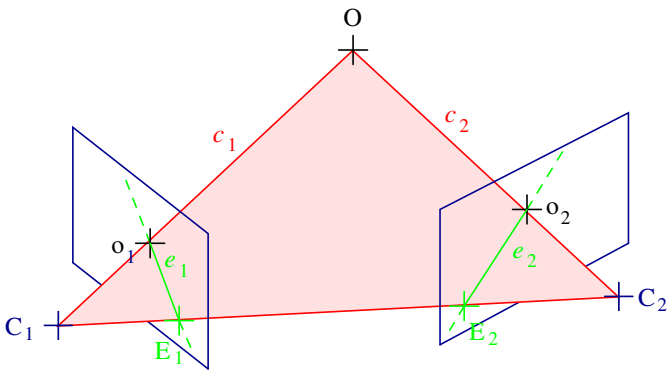


Figure 6.27: **Epipolar geometry.** $\mathbf{o}_1$ is the projection of an unknown real world point $\mathbf{O}$ onto the image plane of camera 1. All possible real world points that can have produced $\mathbf{o}_1$ lie on the ray c_1 from the focal center $\mathbf{C}_1$ of camera 1 through the image point $\mathbf{o}_1$. The points in the image of the second camera that can possibly correspond to $\mathbf{o}_1$ must be located on the image e_2 of c_1 in camera 2. e_2 is called *epipolar line* of point $\mathbf{o}_1$ in the image plane of camera 2. The epipolar line is the intersection of the *epipolar plane* (defined through the focal centers $\mathbf{C}_1$ and $\mathbf{C}_2$, and O) and the image plane of camera 2. The intersection point $\mathbf{E}_2$ of the epipolar plane and the line between the focal centers $\mathbf{C}_1$ and $\mathbf{C}_2$ is the *epipol* of camera 1 with respect to camera 2. The same principles apply analogously to image point $\mathbf{o}_2$ of the second camera.

nically, the left and the right camera pair are treated as separate stereo pairs. In a first step, the positions of visible markers are triangulated in each stereo pair separately. If a marker position is reconstructed from both stereo pairs, its position in 3D space is averaged.

Currently, each of the 18 markers belonging to a hand position is associated with the correct position label in an interactive procedure. An automatic approach that clusters 3D marker positions into separate hand clusters and assigns the marker labels in each cluster according to the colors of their neighboring markers is also feasible.

For motion reconstruction we limit ourselves to those hand positions in which the three markers on the back of the hand are visible for at least two cameras. Only then are the position and orientation of the hand root fully determined. Our setup is arranged such that this condition is fulfilled for an average of three hand positions around the release point. These hand positions are also the most interesting ones in terms of their motion characteristics since they represent that part of the motion cycle in which the hand and finger movements determine the specific rotation axes and spin rates of different pitch types. Sometimes it is not possible to detect the position of all markers in each reconstructed hand position. This can happen for those pitch types where a finger is required to be ahead of the ball in the release moment such that it is occluded from all cameras.

Visualization of the Hand

For visualization purposes, we use the hand model described in Section 6.2. In order to correctly reconstruct the hand poses from the marker positions, we have to make sure that the model matches the pitcher's hand in size and proportions. To this end we apply the warping algorithm from Section 6.2.3 to create a "personalized" hand model that most closely approximates the pitcher's hand. The warped model is then equipped with markers at the same positions as on the glove, cf. Figure 6.25 (right).
Finally, the personalized hand model is animated using joint rotation parameters that have been computed automatically from the marker positions obtained from the tracking process. This conversion from marker positions to joint rotations proceeds as follows. First, we compute the position and orientation of the back of the hand by aligning the three markers on the back of the (personalized) hand model to the corresponding tracked marker positions using a point set registration scheme [Hor87]. Next, we traverse the (anatomical) hierarchy of the hand model along each finger. For each joint, we compute the rotation angle that minimizes the distance between the position of the next marker along the hand model's hierarchy and its corresponding tracked marker. After traversing each finger up to its tip, all joint rotations are specified. We use key frame interpolation for the joint rotation parameters to compute smooth animations.

6.3.3 Tracking the Ball

Marker-based tracking methods are also employed to determine position and orientation of the ball from the multi-exposure images. We have recorded the same four pitch types as for the hand pose measurements. Again, all pitches were performed as three-quarter deliveries.

Preparation of the Ball

We paint optical markers on the ball to be able to estimate its spatial orientation from multi-exposure images. Four different types of markers are used which differ in color and shape (red square, blue ring, green triangle, black circle), see Figure 6.28 (left). Over the entire surface of the ball, each marker type is used three times. Eight markers are arranged in the ball's equatorial

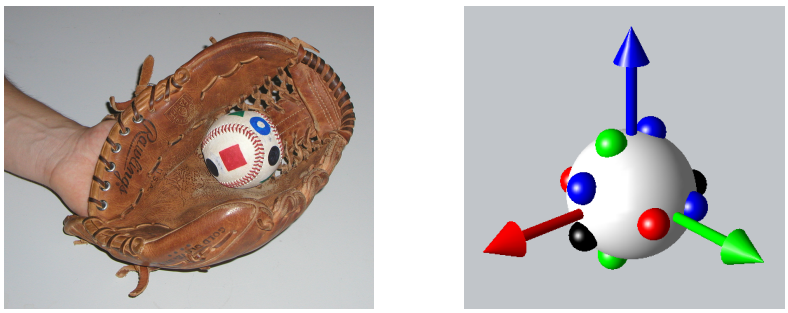


Figure 6.28: **Ball with markers.** Left: baseball equipped with optical markers in pitcher's glove. Right: illustration of the ball's coordinate system. Markers are depicted as small colored spheres on the ball.

plane, in 30°-pairs and with 60° inter-pair separation. The remaining four markers are located in a second, orthogonal plane at 30° distance from the poles. Marker types are assigned such that at least three different markers are visible from any viewpoint. In addition, the (fixed) coordinate system of the ball can be determined from the marker positions for an arbitrary viewing direction (Figure 6.28 (right)).

Acquisition of Ball Motion

To acquire information about the flight of a baseball, two pairs of cameras are used that focus on different aspects of the ball trajectory (Figure 6.22 bottom). The two front cameras take multi-exposure pictures of the first 5 m of the baseball's trajectory right after the ball has left the pitcher's hand. The cameras are placed 3.5 m away from the flight path and are vertically aligned with a baseline of approximately 0.8 m, see Figure 6.24 (left). One strobe light is placed close to the cameras and illuminates the scene such that the ball silhouette appears as a circular shape in the images. In both cameras' multi-exposure images the ball is seen at several subsequent positions and orientations, flying from left to right in Figure 6.29 (left). The number of visible ball positions is determined by the pulse frequency of the stroboscope. At a strobe light frequency of 80 Hz, 6–10 ball positions are captured, depending on the speed of the pitch.
The stereo camera pair in the back part of the setup records the last third of the flight trajectory close to the "home base" where the most interesting variations between different pitches occur. The cameras are placed approximately 2.8 m high and 4 m apart on either side of the flight corridor, see Figure 6.24 (right). A second stroboscope is located below the right camera and illuminates the ball at 50 Hz. This lower frequency is chosen to better separate the ball in the multi-exposure images. In contrast to the camera setup in the front, the illumination direction in the back setup causes partially illuminated ball silhouettes shown in Figure 6.29 (right). We compensate for this before reconstructing ball positions, see below.
During recording, the shutters of the front cameras are open for 1 s, while the shutters of the back cameras expose for 1.3 s. All cameras are triggered simultaneously.
The 3D positions of the ball in flight are recovered via triangulation, see Figure 6.30, and the orientation of the ball's coordinate frame is computed. Then the ball's rotation axis and spin are determined, see Figure 6.30 (left). At 80 Hz, a ball at a spin rate of 1600 rpm rotates by 120° between subsequent strobe flashes. Our sampling frequency is more than twice the spin rate and therefore sufficiently high to fulfill the Nyquist criterion.

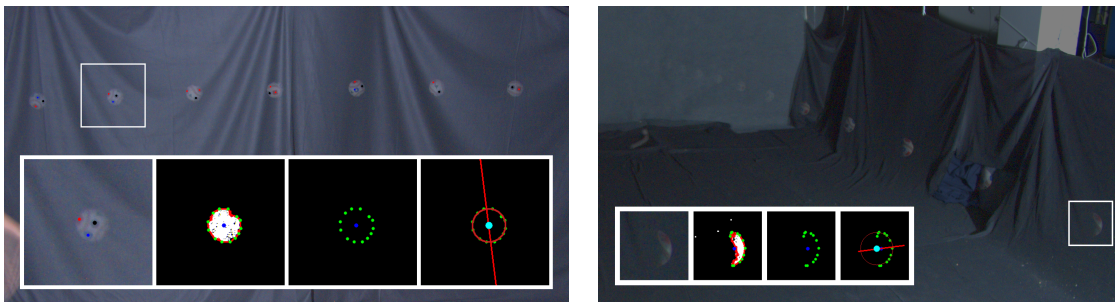


Figure 6.29: **Multi-exposure images of the ball.** Left: multi-exposure image taken by a front camera. Automatically detected markers are shown as colored dots. Right: multi-exposure image taken by one of the back cameras. The half-moon shape of the balls is due to the lateral position of the stroboscope illuminating the flight path. Insets: left to right: magnified image region, result after background subtraction, detected ball silhouette and predicted center point, fitted circle and final center point (see p. 123 ff.).

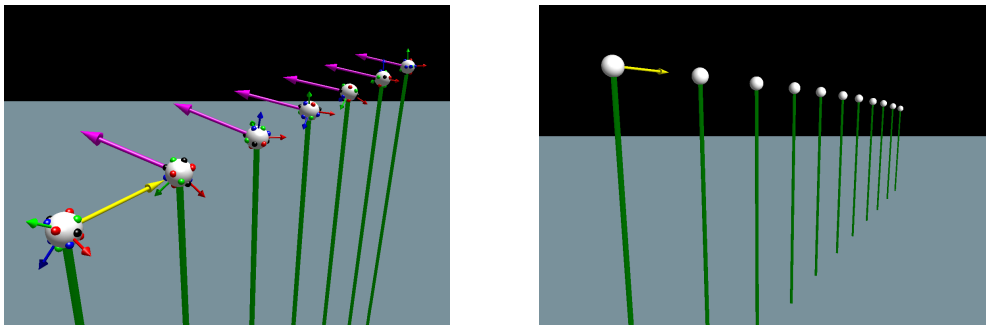


Figure 6.30: **Reconstruction results.** Left: reconstructed initial flight parameters. Right: extracted flight positions of the back cameras. Distance of the balls from the ground is shown in green, rotation axes in magenta, and initial velocity direction in yellow.

Reconstructing Flight Positions

In each multi-exposure image, the silhouettes of the ball in the foreground are separated from the background by means of a color-based background subtraction, thereby creating binary foreground masks. In both the front and back stereo pair of images, the ball silhouettes' boundary polygons are identified via a contour finding algorithm (OpenCV [Int02]). To correct small concavities at the silhouette boundaries of the balls we compute the convex set of the vertices of each boundary polygon [Sla70]. Smaller noise regions are eliminated by imposing a threshold on the region size. First estimates of the projected ball center locations in each image are found via fitting ellipses to the silhouette boundary points. Correspondences between the estimated ball centers in the two images of each stereo pair are established via epipolar lines (cf. Figure 6.27). Approximate 3D ball center locations are obtained by means of triangulation from corresponding ellipse centers for each image pair. From the estimated 3D ball positions and the real-world radius of the baseball, the radius of the reprojected ball is predicted for each camera.
The center estimates in the image planes are further improved by fitting implicit circle models to the silhouette boundaries by means of a Circular Hough Transform (CHT) [Bal81] in a local neighborhood of each ellipse center. The CHT algorithm checks for every sample point (in this case, the vertices of the boundary polygon), and for all possible combinations of circle

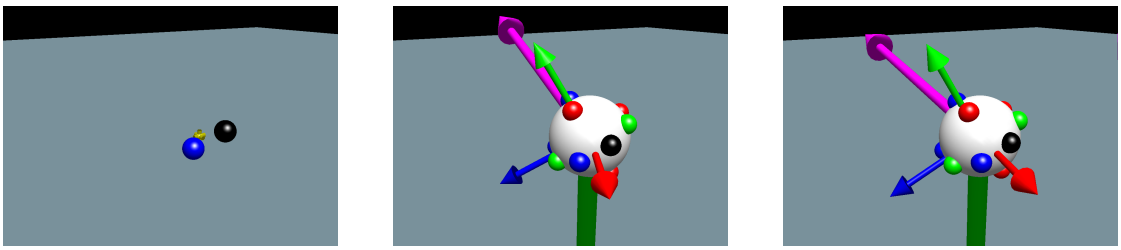


Figure 6.31: **Stages of the fitting process.** Left to right: position of markers, result from prediction, and result from final fitting.

centers and radii within given search intervals, whether the sample point fulfills the implicit circle equation up to a threshold. The circle with the majority of positive votes is the optimal fit. Knowing the radii of the reprojected balls, only the circle centers need to be found, i.e. the CHT search space reduces to two dimensions.
The final 3D positions are found from stereo reconstruction of the circle centers.
The whole fitting pipeline is illustrated in the insets of Figure 6.29. The described procedure robustly recovers three-dimensional ball positions even if the ball silhouettes are only partially visible (Figure 6.29 (right)).

Reconstructing Initial Flight Parameters

After the 3D ball positions in the front and back part of a flight trajectory have been reconstructed, the initial flight parameters for that data set, i.e. velocity, rotation axis, and spin, need to be determined, cf. Figure 6.30 (left). Figure 6.31 gives an overview of the employed technique: from the reconstructed 3D marker positions, an initial guess for the flight parameters is extrapolated, which is then refined using the ball model from Figure 6.28 (right).
The locations of the ball markers are computed in the same way as those of the markers on the pitcher's glove: we identify the projected ball markers in the images of the front stereo pair through color-based region detection and establish correspondences across stereo images via the epipolarity constraint. The markers' 3D positions are found by triangulation, and each marker is assigned to the closest ball position in 3D.
From the sequence of orientations of the ball's coordinate system immediately after release of the ball, its initial spin and rotation axis are derived. In theory, it is sufficient to know the 3D positions of the ball's center and of two uniquely identified markers to determine its orientation. Unfortunately, it is impossible to decide from the color of a marker alone with which of the three instances of this marker type on the ball we are dealing. In addition, misclassifications due to noise in the data need to be considered.
For an ideal flying ball, orientation of the rotation axis and spin are constant over time[2]. Considering the above we determine the initial flight parameters by means of the following numerical optimization scheme.
The algorithm processes the n subsequent ball positions at the beginning of the trajectory separately and in their temporal order. The orientation of the ball at position k with respect to the world coordinate system can be represented as a rotation matrix $R(\alpha_k, \beta_k, \gamma_k)$, where $(\alpha_k, \beta_k, \gamma_k)$ are Euler angles. Our goal is to find for each subsequent pair of 3D ball positions

[2]This does not take into account air friction.

$k-1$ and k the rotation axis $\omega_{k-1,k}$ and rotation angle $\delta_{k-1,k}$ that correspond to the relative transformation $R_{k-1,k}$ between $R(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$ and $R(\alpha_k, \beta_k, \gamma_k)$.
At position k, the algorithm exploits temporal coherence by predicting the orientation of the ball $R(\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}})$ by rotating orientation $R(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$ further by $\delta_{k-2,k-1}$ around axis $\omega_{k-2,k-1}$. Starting from that parameter set $(\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}})$, the algorithm uses Powell's method [PTVF92] to find parameters $(\alpha_k, \beta_k, \gamma_k)$ that minimize the energy function:

$$\begin{aligned} E_k(\alpha_k, \beta_k, \gamma_k) &= w_1 E_1 + w_2 E_2 \\ &= w_1 \sum_{i \in M_k} (\Delta_{\text{marker},k}(i))^2 + w_2 \sum_{a \in \{x,y,z\}} (\Delta_{\text{axis},k}(a))^2 , \end{aligned} \tag{6.10}$$

with w_1 and w_2 being weighting factors. M_k is the set of detected markers at ball position k, $\Delta_{\text{marker},k}(i)$ is the angular distance between reconstructed marker i and the closest marker of the same type in the ball model in the current orientation. $\Delta_{\text{axis},k}(a)$ is the angular distance between the local coordinate axis $a \in \{x, y, z\}$ of the ball in orientation $(\alpha_k, \beta_k, \gamma_k)$ and the same axis in orientation $R(\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}})$.
The rotation axis $\omega_{k-1,k}$ and rotation angle $\delta_{k-1,k}$ are computed from the relative transformation $R_{k-1,k}$ between $R(\alpha_{k-1}, \beta_{k-1}, \gamma_{k-1})$ and $R(\alpha_k, \beta_k, \gamma_k)$ [MLS94].
Having the sequence of rotation angles and the stroboscope frequency f_s, the spin f is computed as

$$f = \frac{1}{f_s n} \cdot \sum_{i=1}^{n} \delta_{i-1,i} . \tag{6.11}$$

In our method we do not strictly enforce the constancy of rotation axis and spin, but instead introduce this criterion as a weighted regularization term E_2 to compensate for possible measurement errors and ball precession. For the initial rotation axis, we average the rotation axes over the sequence. The direction of the initial velocity vector coincides with the direction of the connecting line between the first two ball positions, its magnitude is computed from the strobe frequency and the Euclidean distance of the first two ball positions. For the first two ball positions the optimization is run with $w_2 = 0$ in Equation (6.10). If this initialization fails due to too few or badly located markers, a manual initialization is feasible.
In our experiments we were still able to recover valid initial flight parameters even if for some balls none or just one marker was found. We obtained almost 100 % probability of correct detection for the black markers and 90 % for the red markers. The blue and green markers were more difficult to find due to their similarity in color. In a comparative experiment it turned out that a different color scheme with more luminous marker colors significantly increases the robustness of marker detection.

Validation

For the ball flight data (3D positions and initial parameters), no ground truth information is available. To validate our acquisition setup and tracking algorithms, we show that the data obtained through our measurements and processing are consistent with the prediction of a physically based model that takes into account the dominating forces acting on a spinning ball traveling through air. In accordance to [Ada02] and [AMH01], we compute the velocity $\mathbf{v}(t)$ of a baseball with mass m using the first-order ordinary differential equation

$$m\,\dot{\mathbf{v}}(t) = \mathbf{F}_G + \mathbf{F}_D(\mathbf{v}(t)) + \mathbf{F}_M(\mathbf{v}(t)) \tag{6.12}$$

with the *gravitational force* $\mathbf{F}_G$, the *drag force* (or *air resistance*) $\mathbf{F}_D$, and the *Magnus force* $\mathbf{F}_M$ defined as:

$$\begin{aligned}
\mathbf{F}_G &= m \cdot \mathbf{g} \,, \\
\mathbf{F}_D(\mathbf{v}(t)) &= -\frac{1}{2} \cdot C_D(\mathbf{v}(t)) \cdot \rho \cdot A \cdot |\mathbf{v}(t)|^2 \cdot \frac{\mathbf{v}(t)}{|\mathbf{v}(t)|} \,, \\
\mathbf{F}_M(\mathbf{v}(t)) &= \frac{1}{2} \cdot C_L(\mathbf{v}(t), \omega) \cdot \rho \cdot A \cdot |\mathbf{v}(t)|^2 \cdot \frac{\omega \times \mathbf{v}(t)}{|\omega \times \mathbf{v}(t)|} \,,
\end{aligned}$$

where $\mathbf{g}$ denotes gravity and ρ air density. The cross-sectional area A of a baseball can be found in [Ada02] and [AMH01]. The vector ω represents the spin axis of the ball, which is assumed to be constant during the flight of the ball[3]. To compute the *drag coefficient* $C_D(\mathbf{v}(t))$, we have fitted a polynomial curve to the data presented in [Ada02] and [AMH01]. After computing the Reynold's number $Re(\mathbf{v}(t))$ [Ada02] the drag coefficient is evaluated as

$$\begin{aligned}
C_D(\mathbf{v}(t)) &= 2.23 - \\
&\quad 0.28342 \cdot 10^{-4} \cdot Re(\mathbf{v}(t)) + 0.13179 \cdot 10^{-9} \cdot Re(\mathbf{v}(t))^2 - \\
&\quad 0.25083 \cdot 10^{-15} \cdot Re(\mathbf{v}(t))^3 + 0.17083 \cdot 10^{-21} \cdot Re(\mathbf{v}(t))^4 \,.
\end{aligned}$$

According to [AMH01], the *lift coefficient* C_L can be computed as $C_L(\mathbf{v}(t), \omega) = 1.5 \cdot r \cdot |\omega|/|\mathbf{v}(t)|$. For the special case of a fastball across two or four seams, better approximations of C_L can be obtained from the diagrams in [AMH01]. Given the initial ball position $\mathbf{p}_0 = \mathbf{p}(0)$, the initial velocity $\mathbf{v}_0 = \mathbf{v}(0)$, as well as the initial spin axis ω and frequency $f = |\omega|$, the flying ball's position $\mathbf{p}(t)$ at time t is computed via integrating $\mathbf{v}(t)$ over time. Using the Runge-Kutta-Fehlberg integration scheme `DOPRI5` from [HNW93], we solve ODE (6.12) for $\mathbf{v}(t)$.
Finally, we can compute the reference trajectory of a baseball for a given set of initial flight parameters $\mathbf{p}_0$, $\mathbf{v}_0$, and ω and compare it to our measurements. Since the trajectory computed from the ODE (6.12) is quite sensitive with respect to variations in the initial flight parameters, we search for an exact solution of (6.12) that minimizes the error both for the measured ball positions and for the measured initial flight parameters using Powell's optimization method [PTVF92]. The resulting *optimal reference trajectory* is then used to compute the measurement error (Table 6.6).
The comparatively low average speed of the pitches is due to the high number of pitches per recording session which exceeded the usual training pensum of a baseball professional by far.

6.3.4 Results

For validation of our acquisition setup and tracking algorithms, we have performed a consistency check against a physics-based model of ball flight. As a result, we conclude that our measurements are very accurate. Average errors between the measured 3D ball position and the predicted flight trajectory are as low as 13–19 mm, which corresponds to about 18–25 % of the diameter of the baseball.
The calibration error for the camera setup was on average below one pixel in the image plane. This assures that a high-accuracy 3D reconstruction for the hand markers and the ball is feasible. Due to the lack of ground truth data for the hand motion we cannot assess the reconstructed hand motion data directly. The reprojection errors of the reconstructed marker positions of the hand, however, are similarly small as those obtained for the ball measurements.

[3] For a perfectly homogeneous ball, the spin axis does not change. In practice, a small precession might occur due to the inhomogeneous density of natural materials (cork, leather) used for baseballs.

pitch type	ϵ_{avg}	ϵ_{max}	$\sphericalangle(\mathbf{v}_0^{\text{ref}}, \mathbf{v}_0)$	$\lvert\mathbf{v}_0^{\text{ref}}\rvert$	$\Delta(\lvert\mathbf{v}_0^{\text{ref}}\rvert, \lvert\mathbf{v}_0\rvert)$	$\sphericalangle(\omega^{\text{ref}}, \omega)$	$\lvert\omega^{\text{ref}}\rvert$	$\Delta(\lvert\omega^{\text{ref}}\rvert, \lvert\omega\rvert)$
fastball (2 seams)	18 mm	39 mm	1.3°	63.2 mph	1.9 mph	0.4°	1596 rpm	22 rpm
fastball (4 seams)	18 mm	41 mm	2.5°	64.2 mph	0.8 mph	0.1°	1612 rpm	17 rpm
curveball	19 mm	39 mm	0.7°	61.9 mph	1.4 mph	0.3°	1623 rpm	7 rpm
slider	15 mm	25 mm	3.8°	65.7 mph	0.7 mph	0.4°	1491 rpm	13 rpm
change-up	13 mm	35 mm	1.4°	60.6 mph	1.1 mph	0.3°	1258 rpm	32 rpm

Table 6.6: **Comparison of our measurements with reference trajectories obtained from a physically based model.** For a variety of pitches, the average error ϵ_{avg} and the maximum error ϵ_{max} between the reference trajectory and our measured ball positions are given (Euclidean distance between trajectory and center of ball). The precision of our measured initial flight parameters is specified by: $\sphericalangle(\mathbf{v}_0^{\text{ref}}, \mathbf{v}_0)$ (angle between reference and measured velocity direction), $\Delta(\lvert\mathbf{v}_0^{\text{ref}}\rvert, \lvert\mathbf{v}_0\rvert)$ (difference between reference and measured initial speed), $\sphericalangle(\omega^{\text{ref}}, \omega)$ (angle between reference and measured spin axis direction), and $\Delta(\lvert\omega^{\text{ref}}\rvert, \lvert\omega\rvert)$ (difference between reference and measured spin frequency). Absolute values of reference initial speed $\lvert\mathbf{v}_0^{\text{ref}}\rvert$ and spin frequency $\lvert\omega^{\text{ref}}\rvert$ are given for the sake of completeness.

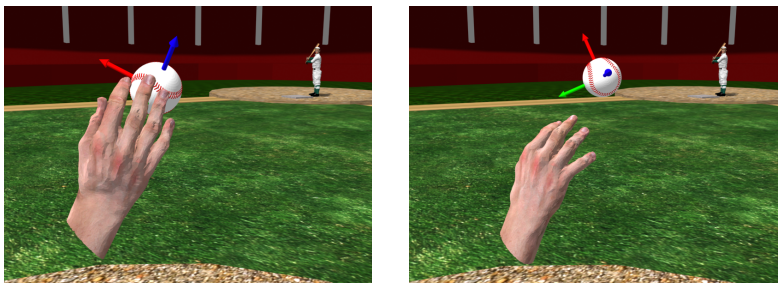


Figure 6.32: **Using the hand for visualization.** Visualization of hand and fingers during and after release of the ball. In this change-up pitch, the ball is spinning backwards about a rotation axis orthogonal to the flight direction. This can be seen by comparing the direction of the axes of the ball's local coordinate frame.

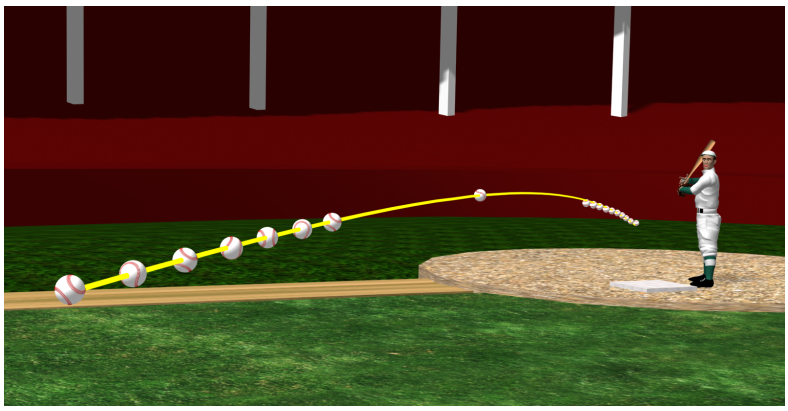


Figure 6.33: **Visualizing the ball trajectory.** Visualization of a change-up trajectory in a stadium. The yellow path shows the reference trajectory obtained from the physical model of ball flight. Balls at the ends of the trajectory are tracked, while the one in the center simulates the flight.

For the ball, the average distance between a measured feature in the image plane and its reprojected 3D location is below two pixels. The reprojection error for the center of the ball is about one pixel. Part of the deviation between measured and predicted ball positions might result from small inaccuracies in feature localization in the image plane.
The high-quality data we acquired from different baseball pitches permit new ways of visualization that provide interesting feedback to the athlete, the coach, and the sports enthusiast. Visualizing the hand motion during release of the ball in slow motion provides a new type of visual feedback for the performing pitcher. Figure 6.32 depicts two snapshots of such an animation. Autodesk® 3D Studio MAX® was used to texture and render the three-dimensional hand model.The flight of the baseball can be visualized from any camera perspective, see Figure 6.33. In particular, the ball's initial flight parameters and their relation to the flight trajectory can be rendered into instructive movies.
The multi-exposure images acquired for tracking the hand motion show both the hand poses and the ball markers. We have thus reconstructed hand motion and flight parameters from the same set of stroboscope photographs. In this way it is possible to visualize the influence of finger motion on the flight parameters of the ball.
In Figure 6.34, the characteristic finger motion applied to add the necessary spin to a slider is

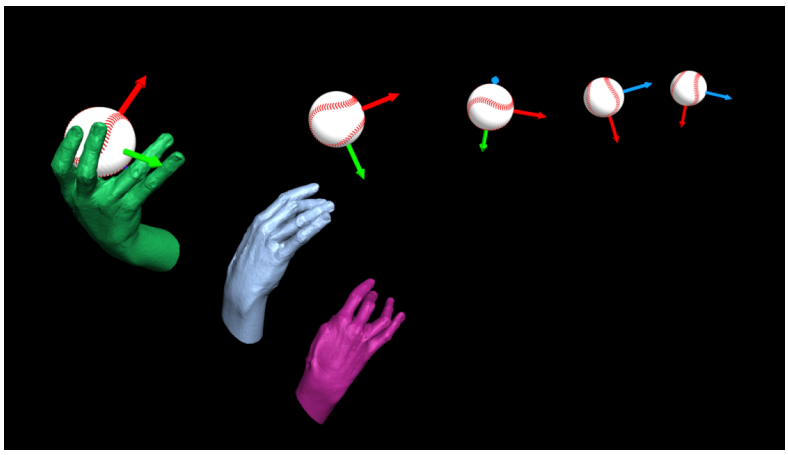


Figure 6.34: **Reconstructed hand poses.** The hand motion during release of the baseball is captured and shown together with the resulting flight characteristics of the ball.

clearly visible. In particular, the middle finger exerts high pressure on the ball to build up high spin. Due to the acceleration of the middle finger during the pitch, this finger moves further than the other fingers after release of the ball. The rotation of the ball in Figure 6.34 is consistent with the movement of the fingers.

6.3.5 Conclusions

We have introduced a setup to capture high-speed, large scale motion via stroboscope photography using off-the-shelf digital still cameras. This passive optical acquisition system permits the reconstruction of complex and fast articulated hand motion. We have shown that the captured motion can be visualized accurately and in detail with the hand model from Section 6.2. In addition, a method for automatic reconstruction of the 3D positions and initial flight parameters of a baseball from multi-exposure images was described and validated.
Our system provides comprehensive and precise measurements of both pitching motion and flight parameters for a variety of baseball pitches. In combination with our visualization techniques, these measurements lead to a better understanding of the characteristics of baseball pitches and resulting flight trajectories. Thus, the system provides an instructive tool for pitchers and coaches, enabling them to improve their pitching technique through precise visual feedback. The demand of such visual feedback was confirmed during our recordings, when we experienced that the athlete's personal estimate of his performance sometimes deviates from the measured data. In particular, assessing the rotation axis correctly seems to be difficult.
We aimed at a high accuracy system that can be used to analyze high-speed motion on a limited spatial and temporal scale. We do not see our approach as a replacement for traditional motion capture techniques, but as a cost effective supplement which can be used in cases where traditional techniques fail. The baseball pitch is just an example for the application scenarios we have in mind. In conjunction with the hand model, hand motion for sign languages such as ASL or legerdemains could be tracked and visualized. Other possible scenarios are tennis serves or the athlete's motion in several track and field events such as javelin or discus. Tennis players, for instance, would benefit from a precise analysis of the correlation between the movement of their racket, speed and spin of the ball, and the resulting ball trajectory during a serve.
Having demonstrated how well the system performs for the difficult problem of hand tracking,

we believe that our framework can be easily extended to capture human full-body motion by means of stroboscope photography.

6.4 Making the Connection: Hand Gestures

This section deals with the generation of conversational hand gestures. By implementing a rule-based approach for gesture generation from input text, the system for non-verbal speech-related facial animation from Section 4.3 and the hand model from this chapter could be combined to yield a character capable of multimodal communication. We sketch how such a component for gesture production might be designed.

Restricting our interest to conversational hand gestures is justified, since they are clearly distinguished from similar movement types, for example from gestures with the intent of object manipulation [QMB+02].

Some researchers believe that gestures are mainly intended to communicate [Bav94, McN92, Ken94], while others [RS91, KCG00] claim their primary function to be to facilitate speaking. The two roles, however, are not contradictory: a gesture can be meant to communicate, and at the same time help to retrieve a mental image [dR98] or help with lexical access.
Gestures share the fate of exact wording: they are formed, possibly decoded, and quickly forgotten, since most often they do not make it into longterm memory [Bav94]. The bond between speech and gestures is a tight one. Gestures emphasize, clarify, and complement speech. Sometimes they are able to express content better than language, for example by pointing, and sometimes they give away aspects that the speaker would prefer to keep hidden [McN92]. A certain degree of redundancy serves to make communication more reliable. Since during the production process, speech and gesture emerge from the same idea unit [Ken80, McN92, QMB+02], they are closely related both temporally and semantically. They are equal expressions of the same content, without one modality being subservient to the other. In general, there is one gesture per idea unit [Ken80], but deviations are possible [McN92].
Gestures fulfil a large number of functions. They can express both static (e.g. shape or location) and dynamic (e.g. movement) aspects of an idea. This idea may be concrete or abstract. People gesture when they introduce new elements into the discourse, or to indicate a return to old ones. In addition, gestures serve to reveal relationships within the discourse structure.
Regardless of conversation type, speakers reveal in their gestures what they regard as relevant and salient in the current context [McN92]. Gestures allow the listener to see the images that the content conveyed in speech conjures up in the speaker's mind: "*... in performing gestures, the speaker's hands are no longer just hands, but* symbols." [McN92, p. 105, emphasis in original]. However, the meaning of gestures is largely context dependent, which makes them hard to decode. In a different situation, the same gesture may illustrate a completely different idea. Moreover, gestures are culture specific and idiosyncratic. Strong interpersonal variations were observed in handedness, function, relative placement, and frequency [KCC96, Kip03]. Gesturing varies also with the addressee [Bav94]. Conversation partners will create between themselves a number of gestures with a shared meaning that all participants will take up.
Typically, a speaker adorns about 3/4 of his clauses in narrative discourse with gestures [McN92], while during conversation, he gestures only during roughly 3/7 of all clauses [Kip03]. Listeners are hardly found to gesture.

As with facial expressions, gesture research is a wide and multidisciplinary field. First, some findings and models from this area are related (Sections 6.4.1 to 6.4.9), succeeded by an overview of gesture generating systems in computer science (Section 6.4.10). The chapter closes with some thoughts on how a gesture generation module compatible with our approach to facial animation might be organized (Section 6.4.11) and some general conclusions on the topic (Section 6.4.12).

6.4.1 Classification of Hand Gestures

The main part of this section describes the widely used classification scheme by McNeill [McN92]. An extensive overview of other systems can be found in [RS91]. Percentages of occurrence for storytelling come from McNeill's experiments [McN92], while those for conversation are taken from [Kip03, p. 165].

Iconics. These gestures depict some aspect of the narration, for example an object or an action. To describe a sphere, for instance, the slightly cupped hands are placed next to each other with the palms facing downwards. Then both hands move in a semi-circle, until they meet again, this time in a supine position. This is the least standardized gesture class. Although some stereotypes exist (e.g. distance between hands indicating size), most iconics are invented as needed. Between 40 and 45% of gestures in narrations and about 5% of gestures in conversation belong to this category. Most of them are enacted in front of the speaker's trunk.

Metaphorics. Metaphorics are pictographic like iconic gestures, but the object or shape they describe represents an abstract idea. The *conduit* metaphor, for example, presents a concept such as knowledge, language, meaning, etc. as a filled container or a substance, which is moved along a path. Often, it is offered to the listener. For example, when talking about a movie, one might say "it was a horror movie" and at the same time perform a gesture where the hands seem to pass a box-like object to the listener. This object does not stand for the concrete, individual movie, but for the abstract genre of horror movies. Metaphorics are rare in narrations. They account for only 7% of cases during storytelling, while with 30 to 40% they constitute a much larger part during conversations. This gesture type tends to occur at belly button height, with rather even horizontal distribution.

Beats / Batons. The hands move with the rhythm of speech, but synchrony is not perfect [McC94]. Regardless of the context, beats always have the same shape. In contrast to other types of gestures, which go through preparation, stroke, and retraction, batons are bi-phasic: an upwards movement succeeded by a downwards movement, or an inwards movement followed by an outwards movement. Beats underline the importance of a word for the discourse structure, e.g. introduction of a new theme or character, or offer the turn to the listener, but they do not illustrate content. They can be superimposed on metaphorics or iconics to signal that the coexpressive words should be considered in the context of the gesture image. In that case, the pose of the illustrative gesture is sustained, and a beat is overlaid. Beats amount to roughly 45% of all gestures in narratives, and to 5 to 15% in conversations. Every person has a favorite spatial location where he executes his beats.

Deictics. Deictics are pointing gestures to concrete or abstract objects or locations. The target of the pointing lies rarely in the real world, but rather in the section of gesture space that

is associated with the object, concept, time or event in question (see Section 6.4.4). Added lateral sweeps indicate plural [Bir71]. An instance of an abstract deictic is the gesturing that accompanies statements of the form "on one hand – on the other hand". In conversations, deictic gestures are frequently used during search for a topic [McN92], but also to assign turns [Kip03]. Pointing gestures are not limited to the hands. The head, for example, is also used to point into a general direction. Deictic hand gestures rarely occur in front of the center of the body, but otherwise are rather unrestricted. They account for 5% of gestures in storytelling, and for 10% of conversational gestures.

Kendon subsumes the above gesture classes under the term *gesticulation* [Ken86], and McNeill's notion of gesture is restricted to them. For these categories, the presence of speech is obligatory. The remaining ones are not subject to this restriction. In fact, the first two of the following classes usually substitute speech and are therefore considered as fundamentally different.

Emblems. As in the case of facial emblems, gesture emblems have fixed form and meaning. They often replace words, as for instance the thumbs-up gesture for "ok". With 30 to 45%, Kipp found them to be the most frequent category during conversation. Since McNeill does not consider emblems to be gestures in the stricter sense, he did not measure emblem frequency during storytelling.

Speech Failures. Often, attempts to recall a word or to find an appropriate sentence structure are accompanied by characteristic gestures like air grasps.

Adaptors. Adaptors [EW69] describe movement that involves touching of self (e.g. scratching one's head) or of objects (playing with a pencil) and that is not immediately discourse related. Therefore, this behavior is in general not considered as gesticulation, although it may communicate aspects of the performer's inner state, like nervousness or boredom. Kipp found adaptors to account for 2% of hand movement during conversation.

The following paragraph introduces another categorization scheme based on different criteria.

Topical and Interactive Gestures. For spontaneous conversation, Bavelas [Bav94] stresses the importance of distinguishing between *topical* and *interactive* gestures. The latter account for 10% to 20% of gestures. They do not refer to the actual content of the conversation but are related to turn taking and serve to incorporate the listener into the interaction without yielding the floor. As such, they usually elicit some listener response. A table listing the different types of interactive gestures can be found in [Bav94, p. 213]. The two classification schemes described in this section do not exclude, but complement, each other.

6.4.2 The Relationship between Speech and Gesture

Gestures are organized into a hierarchy parallel to discourse structure [Bir71, Ken80, McN92] (cf. Table 6.7). At the lowest level, the *accented syllable* corresponds to the *gesture stroke*, i.e. the phase of the gesture that expresses its meaning.
The next unit is the *tone group* or intonation phrase, delimited by pauses and consisting of consecutive syllables that make up a complete intonation tune, e.g. raise-fall. The gestural counterpart is the *gesture phrase*, consisting of *preparation* (optional), *pre-stroke hold* (optional),

kinesic hierarchy	**phonological hierarchy**
consistent arm use and body posture	locution cluster
consistent head movement	locution group
gesture unit	locution
gesture phrase	tone group
stroke	most prominent syllable

Table 6.7: **Parallelism of gesture and speech.** From highest (top) to lowest (bottom) level. After [McN92].

stroke, *post-stroke hold* (optional), and the equally optional *retraction* phase. During preparation, the speaker brings his hands in a position from which he can execute the gesture easily. This happens usually prior to the onset of the parallel speech part. During the pre-stroke hold, the hands are kept in the preparatory pose, while the speaker waits for the speech to catch up with the gesturing. Similarly, during the post-stroke hold, the hand remains in the end posture of the stroke. Retraction, finally, brings the hand back to a rest position. This last part is often omitted, if successive gestures directly pass into each other. This phenomenon is called *coarticulation*. *Modulation* means adjusting the timing of gestures to achieve synchronization [Kip03].
Several tone groups make a *locution*, usually corresponding to a sentence. At their boundaries, locution groups are separated by pauses. In addition, their begin is marked by increased loudness. The corresponding *gesture unit* comprises all gesture phrases within one flow, i.e. between a limb starting from and returning to a rest pose.
Locution groups combine a series of locutions that share a common phonological feature, such as, for example, the same intonation tune. On the gesture side, they are accompanied by a repetition of the same *head movement*.
At the top, there is the *locution cluster*, delimited by pauses, repeated or repaired phrases, altered pitch and / or voice quality, and a shift in topic. In the gesture channel, transitions between locution clusters are marked by a shift in *body posture* and by differences in *arm movement* (e.g. whether the left or right arm is used for gesturing). In written text, this level would roughly correspond to a paragraph.
Not surprisingly, gesture rate is related to speech rate [KCC96].

In spite of this parallelism, speech and gesture are opposites in other respects [McN92, McN02]:

- the parts of a gesture (e.g. hand shape, location) derive their meaning from the meaning of the gesture as a whole – in contrast to language, where the meaning of the individual words determines the meaning of the sentence.
- gestures are less standardized than speech, they are more idiosyncratic and context dependent.
- gestures (except beats) are created from images, but speech from arbitrary mappings between words and meanings.

McNeill repeatedly stresses the "*tightness of the bond between speech and gesture; they are 'unsplittable'*" [McN02, p. 3].

Synchronization between Speech and Gesture

When considering synchronization between speech and gesture, one must bear in mind that the speech affiliate of a gesture is not necessarily a single word, but may well be a complex phrase, i.e. we deal with synchrony between two time intervals (duration of speech and duration of gesture).
Synchronization between speech and hand gestures is less tight than for speech-related facial expressions, and it depends on gesture type.

Iconics, Metaphorics, Abstract Deictics. Generally, at least part of the gesture precedes the relevant speech part. The preparatory phase of a gesture always slightly anticipates the coexpressive part of the utterance [McN92]. 83% of preparation phases occur during the same clause as the stroke, but apart from that, their point of occurrence does not seem to be prescribed. However, most preparation phases start together with grammatical segments, e.g. start of clause, start of verb / noun / preposition phrase, etc. Morrel-Samuels and Krauss [MSK92] report concrete times for the disparity between gesture onset and begin of the articulation of the lexical affiliate: in their data, the range was 0 to 3.75 s, with a mean of 0.99 s. Gestures were never initiated after the onset of their lexical affiliates. Durations of gestures were larger than anticipation intervals in most cases.
If for some reason the stroke is delayed, a pre-stroke hold is inserted after the preparation. The stroke starts either before the tone unit *nucleus*[4], or just at its onset [Ken80], but never after it. This is known as the *phonological synchrony rule*. The same applies to the peak syllable of intonation and intensity in an intonation group [Nob98]. If primary peak of **F0** and intensity peak do not coincide, the rule applies to the last of the two. Stroke and stressed syllable converge in 3/4 of cases, and in 2/3 of them, the stressed syllable equals the nucleus of the intonational phrase [McC98][5], but a complete synchronization between nucleus or peak syllable and stroke does not exist. Mostly, the stroke occurs during the speech affiliate [McN92], but occasionally the stroke is finished completely before the onset of the significant speech.
It may be followed by a hold to allow the associated part of the utterance to catch up with the gesture.
Usually, there is one gesture per clause [McN92] (50% of cases). If a gesture pertains to several clauses, either the gesture is held in a post-stroke hold until the end of the last clause, or the hands return to a rest position after the first clause. After 70% of all gestures, a rest position is assumed briefly and the next gesture follows immediately. In two thirds of the remaining cases, the gesture is followed by one clause without gesture, and by longer pauses else.
During pauses due to interruption, for example due to word search, the gesture can be continued over the pause, but usually there is no gesticulation during pauses. [McN92] reports that 90% of all strokes occur during the actual articulation of speech, and only 1% to 2% during filled or unfilled pauses. The rest is distributed on false starts and breath pauses. According to Nobe [Nob00], 20% to 30% of preparation onset occurs during pauses.

Kendon [Ken80] found pitch and gestures at the end of an utterance to be related as follows: both the gesticulating limb and the final tune are either lowered, or they are both held or raised. McClave [McC98], however, found that the coordination of pitch and movement direction is an

[4]The *nucleus* is the last stressed syllable with a significant change in pitch in an intonation group [Gus86]. This is not necessarily the syllable carrying the peak accent.

[5]The synchrony of gesture stroke and speech accent is used as a cue for gesture recognition by Kettebekov et al. [KYS03].

optional stylistic device, presumably in order to emphasize the corresponding part of the utterance. Its probability depends on the hand with which the gesture is performed. If coordinated, it is the direction of the stroke which mirrors the pitch. For beats, no correlation at all between pitch and hand movement direction existed in her corpus.

Concrete Deictics. De Ruiter [dR98] suggests that synchronization between pointing gestures and their speech affiliates is more tight than for iconics if the gesture is crucial for the understanding of the utterance. "Put that there", for instance, is only understandable if accompanied by pointing gestures. The phonological synchrony rule also applies to deictics.
De Ruiter [dR98] reports that the greatest excursion of the pointing hand (the *apex* of the deictic) occurs on average shortly after the onset of the noun in two-word deictic sentences of the form "*definite article* + *noun*". In this case, speech waits for gesture. For noun phrases of the form "*definite article* + *adjective* + *noun*" (e.g. "the green crocodile" as opposed to "the green lizard" or "the red crocodile"), the start of the pointing gesture is correlated with the position of the word carrying the contrastive stress. For example, the speaker would initiate his deictic gesture earlier when pointing out the *red* crocodile as opposed to the *green* crocodile than when contrasting the green *crocodile* with the green *lizard*. The time between gesture onset and apex depends on the position of the stressed syllable, not only on the contrastive word. As a consequence of these two phenomena, the apex begins later for later contrastive stress, but, in accordance to [Ken80], never occurs after the stressed syllable. Apex duration (i.e. post-stroke hold duration) also increases with stress location.

Beats. According to McClave [McC94], beats do not necessarily occur with stressed syllables. She postulates the *rhythm hypothesis*, which states that beats have a rhythmical pattern of their own, i.e. do not depend on speech rhythm. However, the beat pattern and the vocal rhythm touch at certain points.
The *downpoint* of a beat, i.e. its maximum downward excursion, occurs frequently during unstressed syllables or even during pauses. Intervals between beats are roughly the same within an utterance, but otherwise exhibit great variability: McClave gives examples of utterances with 1/5 s and 1 s periodicity.
People do not always gesture, but if a beat coincides with the nucleus of a tone group, then in the majority of cases the downbeat co-occurs with the nuclear stress. If the tone unit nucleus belongs to a multisyllabic word, the downbeat will start during the stressed syllable, but the downpoint is often not reached before the end of the unstressed syllable following the nucleus. Since more than one beat is possible per word, beats may also occur on unstressed syllables of multisyllabic words.
This alignment of downbeat and stressed syllable applies to nucleic stress only. With monosyllabic words unequal to the nucleus that coincide with a beat, either upward or downward movement can co-occur. In multisyllabic words, however, the downbeat is usually executed during the syllable carrying the primary lexical stress, even if the nucleus is not contained in the word. To summarize, if a beat coincides with the nucleus or primary stress in multisyllabic words, the downpoint can be predicted, but derivation of a beat pattern from stress, word classes, or vocalization is not possible.

It would be interesting and useful to employ gestures of virtual characters for experiments on human sensitivity to speech-gesture (mis-)synchronization. This might even lead to the discovery of a method for beat placement.

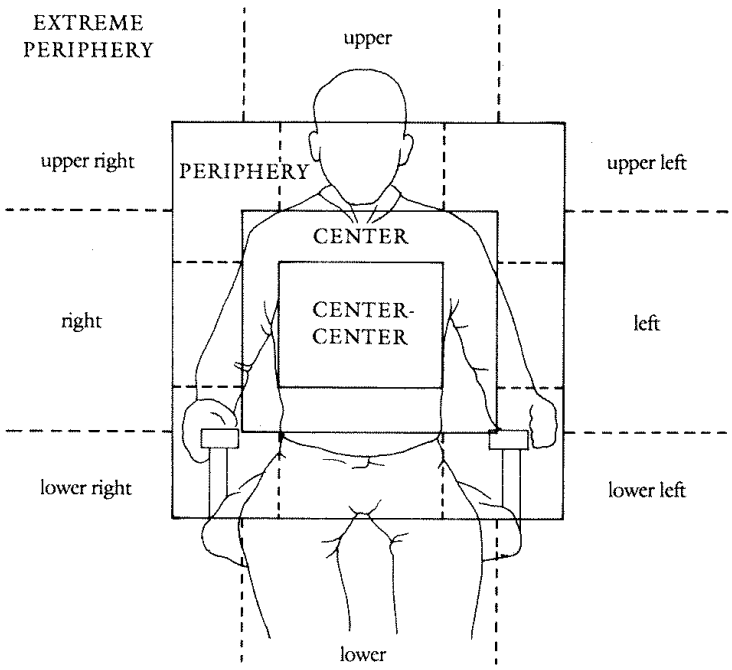


Figure 6.35: **Gesture space.** The space in front of the speaker where gestures occur is divided into several sectors for description purposes. Along the sagittal axis (forward-backward), the spaces is divided in three partitions, which makes a total of 54 sectors. Source: [McN92].

Speech Errors. In general, speech adapts to gesture [dR98]. However, the opposite is true for speech errors. In that case, a hold or, in the case of repetitive gestures, more repetitions are inserted into the gesture stream, or the onset of the gesture is delayed and the preparation phase prolonged, until the normal timing is (almost) restored.

6.4.3 Handedness

The predominant hand in gesturing depends on handedness [Ken80, Kip03]: right-handed persons mostly gesture with their right hand, while hand use of left-handers is more balanced due to a more bilateral organization of speech. For self adaptors, no such difference exists: both hands are employed equally often. In addition, choice of hand is used to mark context coherence [QMB+02], see Section 6.4.5. Kipp [Kip03] observed that handedness also depends on the relative position of speaker and listener (e.g. whether the speaker sits to the right or left of the listener). Bi-handed gestures can either only be performed with two hands, or bi-handedness is used to accentuate the gesture.

6.4.4 Gesture Space

The gesture space [McN92] unites the locations in space where gestures are performed. For description purposes, it is divided into several parts. Figure 6.35 shows a two-dimensional illustration of a sitting person's gesture space. The third, forward–backward, dimension is tripartite. Use of the gesture space is culture dependent: members of different cultures prefer different sectors for different gestures.

Locations in gesture space can be associated with certain events, ideas, or real world places, for example with a particular character. When a reference to this character is made, the speaker points to the corresponding area.

6.4.5 Gestures and Discourse Structure

In storytelling, gestures reflect discourse structure [McN92]: they attribute speech segments to the narrative (i.e. directly related to the story line), the meta-narrative (about the story, for example, “I’m going to tell you a fairy tale”), and the para-narrative (involving the listener, e.g. ”I’m sure you know this type of movie”) level. They can indicate succession, voice (character vs. narrator), perspective (where the observer stands), and distance. If a gesture is held for a longer time than required to convey information, it becomes a question [Bav94]. Hand gestures held at the end of an utterance, however, were also hypothesized to prevent interruption [AC76].
Discourse structure is marked by handedness and (a-)symmetry of two-handed gestures [QMB+02], i.e. speech segments that pertain to the same topic but that are temporally separated are marked by gestures performed with the same hand or the same two-handed symmetry. Form or position are also often shared [McN92]. If, for example, a person briefly deviates from the main theme of her narration in order to explain some detail, a gesture from the main part is taken up again when she returns to the central topic. Recurrent gesture features for the same or closely related topics are due to the similarity of the underlying mental image. The uniting semantic concept behind that image is called *catchment* [McN02, QMB+02].
Lists as a special form of discourse relation can be regarded as catchments. Kipp [Kip03] gives probabilistic relations for gesture occurrence and equality for list items from one of the speakers he observed. If the respective previous list item was accompanied by a gesture, probability for gesturing on the second item was 84%, on the third item it was 67%, and on the fourth item (if existent), even 100%. Chances that a gesture is the same as that of the preceding list item are 68% for the second, 38% for the third, and 67% for the fourth item.

6.4.6 Repetition of Gestures

In general, when a gesture is repeated, the second gesture differs slightly from the first one [McN92]. Either it is more pronounced, for example to highlight a contrast, or less strong. Another possibility is that one hand holds a gesture in order to emphasize a continuing aspect of the recount or some scene characteristic, while the other continues to gesticulate. This contemporaneity strongly connects the gestures and accompanying statements.

6.4.7 Gesture and Emotion

According to Ekman [Ekm65], body movement conveys mainly the intensity of affect and gives little information on the type. He uses a scale to measure the level of arousal that ranges from sleep to tension. When people are relaxed, they do not show their hands or rest them against leg or trunk. With arousal increasing, the hands move in front of the person, but not in space, and with maximal tension they are brought out in space. In [Wal98], Wallbott puts differences in body movement between emotions down to differences at the activation level. However, he also found that certain distinctive features allow the identification of specific emotions from posture and movement characteristics.

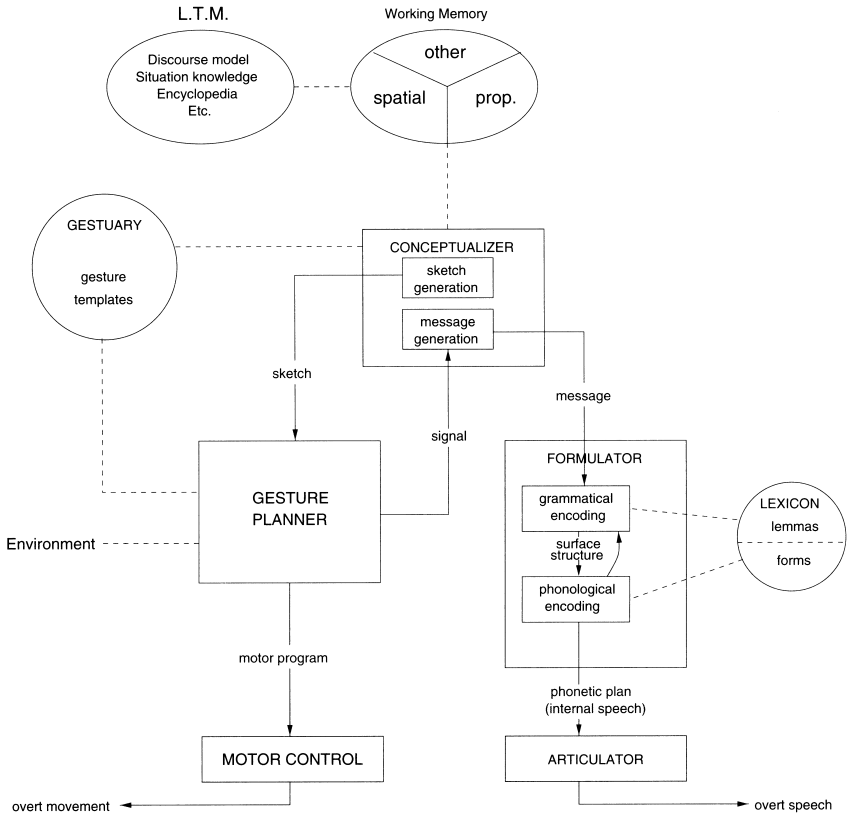


Figure 6.36: **Sketch Model for gesture and speech production.** See Section 6.4.8 for details. Source: [dR98].

6.4.8 De Ruiter's Sketch Model

In his dissertation [dR98][6], de Ruiter proposes an integrated information processing model for the production of gesture and speech. It builds on Levelt's model for speech production [Lev89] and considers all of McNeill's gesture types except beats. Metaphorics are treated as a subclass of iconics. Gesturing is assumed to have communicative intent.
Upon such an intention, the model's *conceptualizer* retrieves the necessary information from a knowledge base. It decides which part of the information is best conveyed by speech, and which by gesture. Accordingly, a *sketch* is sent to the *gesture planner*, and a *preverbal message* to the speech *formulator*, but only after the gesture planner has signaled the completed construction of a motor program (see below). This scheduling leads to a rough synchronization between gesture and speech. In order to create a sketch for certain gestures, the conceptualizer accesses the *gestuary*, a collection of conventionalized gestures. Depending on the type of gesture, the sketch contains information as listed in Table 6.8. For an iconic gesture, one or more spatio-temporal trajectories and the relative position of speaker and gesture are given. Deictic gestures require a vector that indicates the pointing direction and a reference to the manner of pointing (which is culture-specific) in the gestuary. Emblems are stored in the gestuary as a whole, therefore only a reference to the appropriate entry is needed. Pantomimic gestures are determined by a reference

[6]The relevant chapter from the dissertation was also published as [dR00].

gesture type	sketch content
iconic	one or more spatio-temporal trajectories location of speaker relative to trajectory
deictic	vector reference to gestuary
emblem	reference to gestuary
pantomime	reference to motor action schema

Table 6.8: **Sketch composition.** A sketch is the earliest specification of a gesture. Its content varies with gesture type. The gestuary is a knowledge base of conventionalized gesture shapes. See also Section 6.4.8. Source: [dR98].

to a motor program. The sketch does not specify the exact timing of the gesture.
Let us first look at the language generation process and then turn to gestures. The formulator generates a grammatical *surface structure* for the message, which is then encoded phonologically to yield a *phonetic plan*. This is converted into *overt speech* through *articulators*.
On the gesture side, the sketch is transformed into a *motor program* in the *gesture planner*. If appropriate, the pointers to the gestuary or to motor action schemata are dereferenced, and collision avoidance is performed by taking into account the speaker's environment. Gestures are stored in the gestuary as *templates* instead of as fully specified motor plans, because both deictics and emblems have unbound DOFs, for example direction for pointing gestures and place of execution for emblems. The free parameters are filled in during motor program generation. The gesture planner must also handle *fusions* of different gestures. De Ruiter cites the example of a person enacting a throwing gesture while talking about a film he had seen. However, he did not aim at the same direction as the character who threw the ball in the movie, but at the direction from his, the onlooker's, point of view. De Ruiter interprets this as a pointing gesture added to a pantomime. In the case of a fusion, the unbound parameters of the gestuary entry or motor action schema of one gesture are filled in by the second gesture. The gesture planner is also responsible for body part allocation, i.e. decides which body part will execute the gesture.
The *motor control* units, finally, convert the motor program into *overt movement*.

Synchronization via Holds

The sketch is sent to the gesture planner before the initialization of the preverbal message, thus allowing the gesture planner to prepare the gesture and to instruct the motor control unit to initiate a pre-stroke hold. When the preverbal message is sent to the formulator, the gesture planner receives a resume signal and sends the remainder of the motor program to the motor control unit to perform the stroke.
The conceptualizer does not send a retract signal to the gesture planner before completion of the preverbal message, resulting in a post-stroke hold. In case of repetitive gestures, the post-stroke hold is replaced by repeats of the gesture stroke, i.e. the motor program is specified as a loop.

6.4.9 Lexical Retrieval Model of Gesture Production

Krauss et al. [KCC96, KCG00] also developed a gesture production model based on Levelt's model for speech production [Lev89]. However, unlike de Ruiter who believes gestures to com-

municate and to facilitate concept formation during speech production, they started from the assumption that the primary task of gesturing is to facilitate lexical retrieval for speech production. This difference in supposition leads to fundamental differences in the models.
The underlying idea of the model by Krauss et al. is that concepts are stored in memory in several different representations, e.g. propositional (expressed in words) and spatiodynamic (expressed through gestures), and that access of one modality facilitates access of all others. Gesture planning is not actively initiated by the conceptualizer, but by accessing the working memory due to communicative intent. The formulator gets feedback about the enacted gestures in order to benefit from them during lexical retrieval. When the speech has been produced, the gesture is stopped through auditory feedback.

6.4.10 Hand Gestures in Computer Science

Approaches to Gesture Generation

Cassell et al. [CSB+94, CPB+94] developed an inter-agent dialogue system that generates audio-visual speech, intonation, facial expression, and gesture. Starting from the information structure (new / old information), intonation, facial expressions, and gestures are created based on rules. Depending on type and meaning, gestures are instantiated from a library of predefined hand shapes. An additional parameter controls laxness of hand pose. Gesture stroke is synchronized to the accented syllable of the coexpressive word. If a preparatory phase is required, it starts at the latest at the beginning of the associated phrase. The end of the retraction phase coincides with the phrase end. Several gestures within the same utterance are executed without intermediate returns to rest poses, thereby achieving coarticulation between gestures. A final relaxation is always present. Control is implemented via finite state automata.
[LG99b, LG99a] present a high-level specification language for sign language gestures. A sign is described by hand pose, hand orientation, and arm movement. The language relies on a discrete description of space and on movement decomposition into primitives. In addition, an animation system capable of interpreting the gesture specification language was implemented based on a sensori-motor control model for the hands and arms. Coarticulation is achieved by a weighting function that considers directly neighboring configuration targets.
The Virtual Presenter [NZB00] was designed to provide information by explaining presentation boards. The input consists of text annotated with gestures and presentation material. Gestures are synchronized to the following word in the input text. Automatic gesture generation based on words in the input text is also possible. The focus lies on gestural skills for public speaking. The system also takes into account posture and eye contact with the audience. Gaze is used to call the audience's attention to a specific object. As in [CSB+94, CPB+94], finite state automata control the presenter's actions.
Kopp and Wachsmuth [KW00a, KW00b, KW02, KSW04] propose a generation module for coverbal gestures from annotated text. The input is decomposed into coexpressive intonation and gesture phrases for conjoined generation. They orient themselves closely on the gesture generation model by de Ruiter [dR98] (see Section 6.4.8). Their *gestuary* contains gesture templates which describe function and spatiotemporal features of the gesture. Possible fields are hand shape, orientation, location, and movement. Hand shape specifications are based on HamNoSys [PLZ+89], an annotation methodology for sign language. Relations between gesture features like simultaneity can be specified. If a gesture is dynamic, it is composed of several segments. Depending on the function specified in the input and on current configuration, the *gesture planner* selects the appropriate template. Synchronization between speech and gesture

is achieved as follows. The duration of the pause between consecutive phrases is adjusted if the gesture of the following phrase occurs early and requires a long preparation phase. Stroke and lexical affiliate are executed simultaneously. The gesture planner tells the TTS to set the primary stress of the intonation phrase to be within the lexical affiliate. The gesture stroke precedes the onset of the coexpressive speech segment by the approximate duration of one syllable. A rough spatiotemporal plan for the strokes of consecutive gestures is made, which is subsequently refined by the *motor planner*. After each gesture, a stereotyped retraction movement is appended with slight overshooting before coming to rest. Transitions between consecutive gestures take coarticulation into account. The final animation is obtained by a kinematic approach that considers natural dynamics.

The rule-based Behavior Expression Animation Toolkit (BEAT) [CVB01] derives intonation, lip sync, facial expressions, and gestures from input text based on a linguistic and contextual analysis and on knowledge bases. Gestures are proposed only for new or contrastive information in a clause. The system takes into account that people tend to gesture about unusual aspects of objects, even if these are not mentioned verbally. First, all possible gestures are identified for each utterance. This set is then reduced through user-defined filters to only display those movements appropriate for a certain character. Conflict resolution is priority-based. Coarticulation is here understood and implemented as the superposition of one gesture over another which uses the same DOFs. Duration of gestures are not modified. [CNB+01] extends the system to incorporate posture shifts.

In [HMP02], a language for gesture description is proposed as well as an animation system for gesture synthesis. Each gesture is described by a series of keyframe templates containing fields for hand shape, wrist orientation, arm movement, and place of articulation. Unspecified properties are determined by interpolation. HamNoSys [PLZ+89] is used to describe hand pose and orientation, and position is specified within McNeill's gesture space [McN92] (Section 6.4.4). The exact timing of a frame depends on the neighboring gestures. In the synthesis system, a gesture planner instantiates each gesture present in the input XML structure, inserts rest positions as necessary, or automatically overlays beats over other gestures in case of multiple intonation peaks in a clause providing new information. Follow-through[7] for the arm is also implemented. The succession of gestures is then translated into joint angles depending on the human model.

Krenn and Pirker [KP04] designed a system independent gesture knowledge base, the *gesticon*. Each gesticon entry contains information about gesture type, shape, meaning, coexpressive speech segment, type of alignment to the segment and relative timing, i.e. position and dynamics (separately for preparation, stroke, hold, retraction). Additional constraints can be added, for example, to restrict a certain gesture to particular emotions. The gesticon is used in the following system: the scene generation and affective reasoning components pass dialog acts to the multimodal natural language generator which creates a text representation and assigns posture shifts, hand gestures, and facial expressions to speech segments based on semantic and pragmatic content. The text is transformed into emotional speech by the MARY TTS [Sch04b, ST03] (see Sections 4.3.2 and 4.4.2). The prosodic and timing information from the TTS is used to align speech and non-verbal movement. Time permitting, a return movement to a rest position is inserted between gestures.

Stone et al. [SDO+04] use pre-recorded chunks of audio and chunks of movement to synthesize new full-body utterances. Possible sentences the character can say are described by a grammar. From this, the sentences that need to be recorded to obtain the speech and gesture chunks

[7] Movement is propagated from the body core outwards. Consequently, there is a small delay between movement of consecutive joints in the hierarchy.

required to form all possible utterances are determined. For synthesis, a sentence is generated according to context. The corresponding utterance is assembled from those speech phrases and gestures that match the communicative function and minimize the required amount of time warping and blending.

Automatic Derivation of Iconics from Object Descriptions

Both [KSW04] and [KTC04] deal with automatic generation of iconic gestures from object descriptions and site plans. [KSW04] approaches the problem from two directions. On the one hand, the strokes of motion captured gestures are automatically expressed in a gesture description language and subsequently imitated by an embodied agent. On the other hand, a description language for objects is introduced that captures the most salient features such as the direction of maximal diameter or rounded shape. From such a description, iconic gestures can be generated automatically, because it is these dimensional characteristics that humans encode. Conversely, the language can also be used for the description of object features from gestures. Putting these two approaches together would allow the representation of tracked gestures in the gesture description language to be mapped to the object description language, which would permit the generation of new, possibly different gestures from this representation, i.e. the agent could rephrase the user's gesture.
The system described in [KTC04] derives natural language and gesture from the same communicative concept. For gesture formation, the imagistic content to be conveyed by the gesture is broken up into *image description features*, which are linked to discrete form features like hand shape or trajectory. During utterance composition, the *gesture planner* generates all possible gestures from the image descriptions. These are collected in an add hoc lexicon of gestures from which the natural language generator selects the one that, together with the generated language, expresses the content best. Gesture and speech affiliate are paired for synchronization. The scheduling and motor planning components are from [CVB01] and from [KW04], respectively.

Individual Gesturing Styles

In his dissertation, Kipp [Kip03] describes an empirical approach to gesture generation that models gesturing styles of individuals. From an annotated corpus, *gesture profiles* are derived for every target person. A profile comprises a probabilistic concept-to-gesture mapping (e.g. rejection maps to a wiping hand gesture with a certain probability) and statistical models for timing, handedness, transitions, variation, and frequencies of gestures. In the process, a lexicon of the encountered gesture equivalence classes was assembled. The profiles are subsequently used to generate gestural behavior from annotated input text. In a first step, all possible gestures are generated from the profiles. They are then filtered based on the annotations in the input text (segment boundaries, morphological information, new / old information, focus, and discourse relations (lists, opposition, repetition)) and on the gesture profiles. Collisions between gestures are resolved by selecting the next likely gesture instead of the one causing the clash. The system outputs an abstract script containing for each gesture a pointer to the gesture lexicon, handedness, relative timing, and speech affiliate. Gestures related to discourse structure that do not have a direct speech affiliate are not considered.
[NR04] is also concerned with individual gesturing style. A *style* consists of one or more *style dictionaries* containing probabilistic mappings of meaning to specific gestures, a *manner definition* for motion characteristics, and a *modality usage* parameter to indicate preference of body

parts (e.g. face or hands). Combining style dictionaries yields mappings for new cultural groups or individuals, for example a style dictionary for teachers together with one for Dutch will result in a style dictionary for Dutch teachers. Gestures are characterized by attributes for intensity, noise, manner, and duration, which may be changed via *style modifiers*. A *gesture repertoire* serves as knowledge base. Input text is annotated with gestures or high-level meaning tags that are automatically translated to style modifiers and gestures based on the current style.

Expressivity of Gestures

Chi et al. [CCZB00] do not synthesize gestures from scratch, but modify existing key frame animations by setting effort and shape parameters of Laban Movement Analysis to obtain expressive animations. Shape influences the expansion of the arm movement by modifying key poses, while effort has an impact on the execution of the movement, i.e. on roundness, interpolation space (end effector position, joint angles, etc.), velocity, and acceleration. For the torso, shape changes essentially result in "squash and stretch"-like movement, for example, by leaning forward or being very erect.
A similar approach was proposed by Hartmann et al. [HMP05] as an extension of [HMP02]. Changes in expressivity are marked in the input text and incorporated at the gesture synthesis stage. Parameters for expressivity are overall activation (modeled as gesturing frequency), spatial extent (movement amplitude), temporal extent (movement duration), fluidity (smoothness and continuity of movement), power (dynamic properties), and repetition (tendency of superimposed beats). The features of a gesture that carry the meaning are not affected by the expressivity parameters.
To a limited extent, Ruttkay et al. [RNtH03] also implemented expressivity for hand gestures by modifying the parameters for precision (noise), dynamism (acceleration / deceleration pattern), and intensity. For each level of intensity, a variation of the gesture (e.g. a different hand shape) must be present in the knowledge base.

6.4.11 Gesture Generation for Speech

In this section, we detail some thoughts about an annotated-text-to-gesture module that would fit well with our approach to facial animation from text as described in Section 4.3. Since it is derived from the model by de Ruiter [dR98] (see Section 6.4.8), it is necessarily similar to other approaches that originated from the same theoretical model (e.g. [KW00a, KW00b]).
We believe that de Ruiter's Sketch Model is a better starting point for our situation than the model by Krauss et al. [KCC96, KCG00] (see Section 6.4.9), since we do not attempt to simulate memory retrieval processes. Furthermore, we start from an integrated representation of text and gesture tags, which is more in line with the Sketch Model.

Figure 6.37 illustrates our method and allows comparison with the de Ruiter model in Figure 6.36. Instead of receiving a communicative intention from the working memory as in the Sketch Model, the *conceptualizer* gets text input with gesture (and possibly emotion) tags from the user. Tags are necessary in order to identify speech affiliates. As preverbal message, the conceptualizer forwards the text (with all tags) to the *formulator*, i.e. the MARY TTS (see Sections 4.3.2 and 4.4.2). The sketch sent to the *gesture planner* has more or less the same form as proposed by de Ruiter: a sequence of gesture templates that depend on gesture type (see Table 6.8), filled in with information from the input annotation.

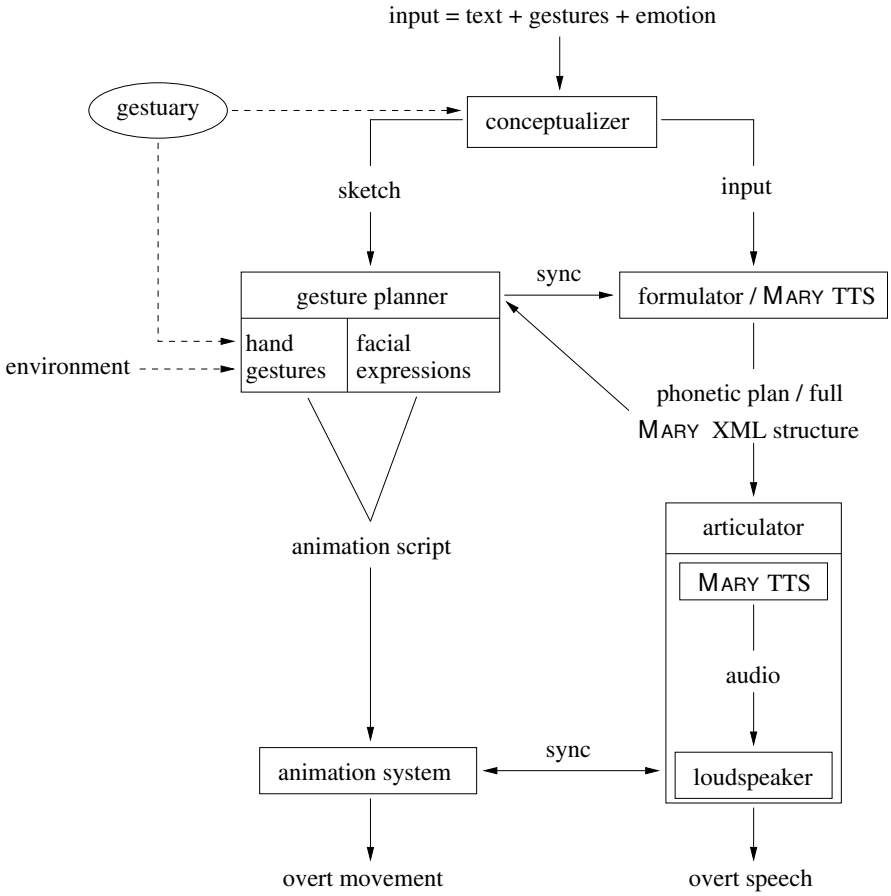


Figure 6.37: **Proposed architecture for gesture generation.** Text input annotated with hand gestures is transformed into speech and animations of facial expressions and gestures. The MARY TTS was introduced in Section 4.3.2. How we generate facial expressions from plain input text is described in Section 4.3. Gesture production follows de Ruiter's model [dR98] (Section 6.4.8) as closely as possible. For a description of the gesture generation module, see Section 6.4.11.

At the next stage, the formulator / TTS performs the phonological encoding of the input text. The resulting MARY XML structure is passed on as *phonetic plan* to the first part of the *articulator*, i.e. the actual synthesis module of the TTS which generates a speech audio file. The second part of the articulator are the PC speakers. The gesture tags are passed through all stages in the TTS and are present in the final XML document, allowing to synchronize gesture and speech.
Unlike in the original Sketch Model, the gesture planner has access to the phonetic plan, which it must parse in order to generate the facial animation (see Section 4.3) and to appropriately synchronize the hand gestures to their lexical affiliates. On the other hand, a path from the gesture planner to the formulator is also necessary, since for elaborate gestures of long duration, speech onset depends on gesture duration (otherwise, correct synchronization can no longer be guaranteed), i.e. the phrase initial pause must possibly be adjusted in the phonetic plan as indicated by the gesture planner.
The gesture planner handles each gesture category differently, see below. It computes the timings of the gestures in the *sketch* from the XML structure of the TTS, dereferences the pointers to the *gestuary*, and is responsible for body part allocation. During the generation of the actual

animation script, it must take into account coarticulation, i.e. blend gestures that directly follow each other without a rest interval. Possible empty fields in the template are either filled by fusing two gestures, if specified in the input, or by interpolation between surrounding gestures.
Both at the gesture and the facial animation level, emotion should be taken into account. For the face, we have described an approach that is compatible with the emotion module within MARY (Section 4.4). In case of gestures, if one of the existing approaches [CCZB00, HMP05] is chosen for integrating emotion and gestures, a mapping from either emotion words or, preferably, the emotion dimensions activation, evaluation, and power into their expressivity space is required.
Together with the specification of the facial animation, the final *motor plan* is passed to the *motor control units* for execution. Currently, we only have animation systems for the face and the hands. Integrating these two high-detail components with a simpler model for the rest of the body would yield animations with high detail in the most important areas and at the same time avoid an explosion of computation time.
Synchronization between audio and animation is inherent due to the shared underlying timing, but the same process must trigger both animation and audio output. Since gesture precedes speech and generally, speech seems to adapt to gesture [dR98], the motor control component would be one candidate to initiate both animation and audio playback, but a better solution from a software design point of view might be the conceptualizer. When the gesture planner has generated the motor program and the TTS the audio file, both components signal to the conceptualizer which activates both animation and audio output and ensures synchrony. Similarly, it might be neater to force gesture planner and formulator to communicate through the conceptualizer during the planning stage instead of directly with one another as indicated in the diagram.

In order to slim down the design, one could also imagine the TTS to directly receive the input, to process it, and to pass its output to the gesture planner, thereby initiating the gesture and facial animation planning stage, during which the hand gesture module generates and fills in the sketch, and finally creates the animation from it. After gestures (with a possible adjustment of the speech timing) and facial animation have been generated, the gesture planner activates the animation system and the audio output. In this way, we would arrive at a linear design.

Gestuary

When populating the gestuary, the gesture collection by Kipp [Kip03] might be a good starting point. He assembled a lexicon of 68 gesture equivalence classes from his corpus, mostly emblems and metaphorics, but also some deictics, iconics, self-adaptors, and beats. It contains for each gesture handedness, hand shape and movement, orientation, location, frequency of occurrence, and a sample of a lexical affiliate. In order to include function, semantic tags must be assigned to the gestures. The difficulty here is that different people use the same gesture with a different meaning. However, this may be exploited to model individual behavior [Kip03].
Kipp argues persuasively that using a lexicon of gesture makes sense. In his conversational corpus, he found iconics to be rare, and metaphorics are to a certain degree standardized. The data from his experiments supports the claim that – apart from iconics that are invented on the fly – people use a shared, finite collection of gestures. From the gesture equivalence classes in his corpus, Kipp distilled a shared lexicon of frequent gestures that covers 85% to 90% of the data. 15 out of the 39 gestures from the lexicon were used by both speakers. They constituted around 55% of the individual repertoire of each speaker.

Gestuary Entries. Entries need to consider handedness, hand pose, orientation, and location, as well as the development of these aspects over time. Handedness is constant during one gesture and can either be right hand, left hand, or both hands. For hand shape, it makes sense to build on the field-tested HamNoSys [PLZ+89]. Orientation cannot be specified as an absolute orientation, like "palm up", because it depends on the character's pose. If it enacts an iconic for newspaper reading while lying on its back, the gesture must be rotated by 90° compared to the same gesture executed while sitting in a chair. One possible frame of reference for hand orientation is the gesture space. Location is best expressed as a sector of the character's gesture space. Within this sector, the exact position can be chosen randomly to introduce variability. Only for concrete deictics, an exact location is required. Temporal development can be achieved by giving key frames, relative timings, and a reference to an interpolation scheme. For example, half-circular sweeping movements need to be interpolated differently than tracing a zig-zag course of straight lines.
It is not necessary to always specify all parameters in a gestuary entry. Values are required only for those fields that carry the meaning of the gesture. These values can be part of either the gestuary entry or of the sketch. Location of the hand for deictics, for example, is derived from the pointing vector. Unspecified parameter values are automatically filled in by the gesture planner by interpolation or by gesture fusion, if specified. Handedness is a special case. Here, the gesture planner should choose the same hand as for the previous gesture, but also consider restrictions from the environment and the character's original handedness. It should be possible to specify this parameter in the input annotation, because handedness plays an important role in marking discourse relations.

Gesture Representation

The representations for the individual gesture categories in the sketch emerge from Table 6.8.

Pantomimes. In the Sketch Model, pantomimes are specified as references to motor action schemes. The analogon in our case are references to the respective animation files, possibly from motion capture for the most natural dynamics. A question in this context is whether pantomimes are always enacted at the same speed, or whether the animations would have to be time scaled. After blending to the preceding and succeeding gesture has been performed, or a transition to a resting pose has been inserted, the animation snippet can be directly integrated into the final animation stream.

Iconic and Metaphoric Gestures, Emblems. For these gesture classes, the sketch contains a reference to the gestuary.

Deictics. Pointing gestures require a pointer to a gestuary entry that consists of a single frame, where the only field specified is hand shape. From the vector in the sketch, orientation and location can be computed.

Beats are not coded explicitly, they are realized as up and down movements with a relaxed hand shape, or the current hand shape, if they are superimposed over other gestures.

So far, most systems have their own gesture specification language. In order to make an exchange possible and thus assemble a larger gestuary, a standard ought to be developed [KP04].

Synchronization

From the literature survey in Section 6.4.2, it became obvious that synchronization to speech depends on gesture type.

Iconics, Metaphorics, Abstract Deictics. For these gesture types, we assume that the user has specified only speech affiliates that do not occur after the nucleus of the intonation phrase (which probably would not make sense, anyway). Otherwise, the gesture planner would have to modify the point of occurrence of the tone unit nucleus in order to prevent a violation of the phonological synchrony rule. We do not model cases where the stroke is finished before the onset of the speech affiliate.
In compliance with [MSK92], we let the preparation phase start as a default at the word boundary that is approximately 0.99 s before the coexpressive speech segment. If the preparation finishes perceptibly before stroke onset when performed at a comfortable speed, the preparation phase is shifted forward in time, but not more than to the beginning of the affiliate, and the stroke is moved backwards, until the two phases meet. Another possibility is to insert a pre-stroke hold.
According to McClave [McC98], 50% of all strokes coincide with the nucleus of the intonational phrase, and another 25% with a stressed syllable that is not the nucleus. Therefore, we decided to use the latter of the peak **F0** syllable and the intensity peak of the speech affiliate as point of reference (see [Nob98] and also [KW02]), provided it precedes or equals the nucleus, and the nucleus otherwise: the gesture stroke is scheduled to end with the reference syllable. If the stroke is so short that it would start after the reference peak, its onset is shifted to this peak. This guarantees that the stroke does not start after the last of intonational and intensity peak syllable, nor after the nucleus, i.e. phonological synchrony is preserved.
Depending on the length of the gesture and of the anticipatory movement, the begin of the preparation phase may have to be shifted backward in time. In case this leads to a collision with the previous gesture, the pause between the two phrases must be extended.
In de Ruiter's model, a post-stroke hold is inserted until the preverbal message has been completed. Since we do not model this process, we can either force a hold (or repetitions, in case of repetitive gestures) until the end of the coexpressive speech, or until its penultimate syllable, since the preverbal message must be completed before the speech. Then, retraction commences. At any time, the preparation phase of the next gesture can interrupt it, which leads to a blend into the anticipatory phase of the succeeding gesture.

Concrete Deictics. For this category, the findings by de Ruiter [dR98, p. 53, p. 64] can be implemented. As the TTS does not know which the contrastive word is, this must be given in the input annotation in order to obtain the desired intonation from the TTS. Since the user must also specify the lexical affiliate, we know the number of coexpressive words. If this is two and if the first word does not contain the nucleus, we propose to set the apex of the pointing gesture to the end of the first phoneme of the second word. In case the first word does contain the nucleus, or if the affiliate only consists of one word, the same timing is applied to the first word.
If the coexpressive speech consists of three words, apex position depends on contrastive stress. If the peak syllable occurs during the first word, we apply the same rule as for two words. In the case of the second word carrying the main accent, the apex is reached after the first syllable of the second word and held until the end of that word. Otherwise, it is set to the beginning of the last syllable of the second word. A post-stroke hold is inserted until the end of the first syllable of the last word.

The worst case is more than three coexpressive words, because here we do not have explicit rules. These pointing gestures must be handled like abstract deictics in order to enforce phonological synchrony.
Launch time seems to be approximately 600–800 ms, depending on stress location. If this interferes with previous gestures, the pause between the two phrases must be elongated.

Beats. Beats are the only gestures that do not stand in any relation to the content of the utterance. According to McClave [McC94], their points of occurrence cannot be predicted. We have two possibilities here: either all beats must be specified in the input, or we attempt to automatically determine a beat pattern in spite of McClave's findings. This would allow us to investigate human perception of possibly wrong beat patterns. The second alternative should definitely be attempted. Maybe it will even lead to a heuristic rule for perceptibly acceptable beat positioning.
In case the beats are given in the input text, we can proceed as follows. The output of the TTS allows us to identify the tone unit nuclei. If a beat was prescribed for the respective word, the downbeat starts and ends with it in case of monosyllabic words. For multisyllabic words, it begins some time during the nucleic syllable and ends during the following syllable. If a beat is specified for a multisyllabic word that does not contain the nucleus, the downwards movement is synchronized in the same way to the lexically stressed syllable. Exact timing within a syllable does not seem to matter, so we can choose random times that guarantee a minimum duration. If a beat is requested for a non-nucleic monosyllabic word or a second beat for a multisyllabic word, its downpoint is scheduled to form a regular pattern with the previous and following gesture, since to some extent also other gesture types than beats form part of the personal rhythm [McC94].
Automatic placement of beats could assign beats to the tone unit nuclei and insert other beats in between in order to obtain a beat rhythm with periodicity roughly between 1/5 s and 1 s, where periodicity may vary up to 1/15 s in either direction and may be halfed or doubled.

A study would have to reveal whether the possibility to specify different gesture-speech alignment schemes in the input annotation (e.g. gesture starts with speech affiliate or gesture finishes before speech affiliate) [Kip03, KP04] is a tool that animators find useful, or whether it complicates matters unnecessarily.

6.4.12 Conclusions

The goal of this section was to show how generation of hand gestures from annotated text and text-based facial animation can be coupled, thus bringing together our approach to coverbal facial animation and our hand animation system.

The most challenging part of gesture planning is synchronization of gesture and speech. The temporal relationship between the two modalities is rather loose. More research in this area is required, since there are still a number of open questions, for instance, concerning beat rhythm. Learning-based animation systems might be of use here, but the training phase would require a lot of annotation work (tone unit nuclei, accented syllables, pitch, lexical affiliate, etc.), since gesture-speech synchronization appears to involve many factors, and success is not guaranteed.
Variations in gesture patterns due to emotions or different personal style are an important tool to create entertaining and believable animations of gestures. Small random variations (e.g. of their exact location) help to avoid robot-like repetitiveness. Especially when having several

agents at the same time that communicate with each other and / or with the user, it is important to implement between-agent variations to endow the virtual characters with individuality, since when and how people gesture varies between and within speakers.

Animations of arm movement must allow for a high level of control and flexibility, while at the same time appear lifelike. Naturalness in existing systems is sought by taking into account, for example, tension, continuity, and bias [HMP02] and Fitts' law[8] [KW02, HMP05]. Kopp and Wachsmuth [KW04] break the movement into segments with bell-shaped velocity profiles. They consider the relationship between velocity and movement shape as well as between shape and duration. Lebourque et al. [LG99b] also assemble arm movement from movement primitives that have natural velocity profiles. Furthermore, movement overshooting is implemented in [KW00a, HMP05].
Animating hands, arms, and face is not enough. One should at least include the torso [CCZB00], preferably employ the entire body and take into account that face and hands / arms are not the only body parts that can perform gestures in the wider sense.
Coordinating hand gestures with other behavior will enhance naturalness and perhaps even communication effectivity. For example, the speaker orients his gaze at his hands to call the listeners attention to the gesture (iconic, metaphoric, or deictic) [Str94].

Gesture generation systems permit to conduct interesting perception experiments that are difficult to conduct with human stimulus material, since people are not able to modify their gesturing behavior completely at will. It would, for example, be a lot easier to construct stimulus material for testing the effect of speech-gesture mismatches with an animation program than by choreographing humans as described in [MCM94]. In addition, one could easily measure how sensitive people are with respect to timing. These and other questions seem to be worth investigating.

[8] Model of human psychomotor behavior for rapid, aimed movement [Fit54].

7

Conclusions

This thesis presents various contributions aimed towards automating and improving the communication skills of virtual humans. In the realm of facial animation, two different methods were proposed to automatically augment animations of lip sync with non-verbal speech-related facial expressions, such as raised eyebrows and head nods on accented syllables, or eye blinks at the end of grammatical clauses to signal speaker continuation. The first technique deduces speech accompanying facial movement directly from an analysis of the (natural) speech signal, while in the second case, the linguistic analysis of a coupled text-to-speech system provides the necessary fundamental information. The text-based method also allows the user to insert emoticons into the text that are then integrated as facial expressions at corresponding positions in the animation. With both approaches, synchrony of animation and audio signal is inherent.
Furthermore, we have presented an algorithm for generating facial expressions for a continuum of pure and mixed emotions of varying intensity. Based on the observation that in natural interaction among humans, shades of emotion are much more frequently encountered than expressions of basic emotions, a method to generate more than Ekman's six basic emotions (joy, anger, fear, sadness, disgust and surprise) is required. To this end, we have adapted the algorithm proposed by Tsapatsoulis et al. [TRK$^+$02] to be applicable to a physics-based facial animation system and a single, integrated emotion model. The facial animation system was combined with an equally flexible and expressive text-to-speech synthesis system, based upon the same emotion model, to form a talking head capable of expressing non-basic emotions of varying intensities.

We have also presented a novel approach to create plausible 3D face models from vague recollections or incomplete descriptions. This task plays an important role in police work, where composite facial images of suspects need to be created from vague descriptions given by eyewitnesses of an incident.
Our approach is based on a morphable model of 3D faces [BV99] and takes into account correlations among facial features based on human anatomy and ethnicity. Using these correlations, unspecified parts of the target face are automatically completed to yield a coherent face model. Through an intuitive GUI, the system provides high-level control of facial attributes as well as the possibility to import facial features from a database. In addition, the user can specify a set of attribute constraints that are used to restrict the target face to a residual subspace. These constraints can also be enforced on the example faces in the database, bringing their appearance closer to the mental image of the user, and thus avoiding confusing exposure to entirely different faces. Adapting the system to local populations is achieved through additional image databases that are converted into 3D representations by automated shape reconstruction.
We have demonstrated the applicability of our system in a simulated forensic scenario and have

compared our results with those obtained by a professional forensic artist using state-of-the-art software for creating composite images in police work.

The hands constitute another important modality of human communication. Here, we have contributed a physics-based human hand model with underlying anatomical structure. Animation of the hand model is controlled by muscle contraction values. We employ a physically based hybrid muscle model to convert these contraction values into movement of skin and bones. Pseudo muscles directly control the rotation of bones based on anatomical data and mechanical laws, while geometric muscles deform the skin tissue using a mass-spring system. Thus, resulting animations automatically exhibit anatomically and physically correct finger movements and skin deformations. In addition, we present a deformation technique to create individual hand models from photographs. A radial basis warping function is set up from the correspondence of feature points and applied to the complete structure of the reference hand model, making the deformed (new) hand model instantly animatable.
The model was used to visualize the hand movement of a baseball pitcher immediately before, during, and after ball release. Hand poses at key positions were obtained from a tracking system based on multi-exposure photography. This approach permits to capture high-speed motion with low-cost commodity still cameras and a stroboscope. The recorded motion remains completely undisturbed by the motion capture process. We acquired the motion of both hand and ball for a variety of baseball pitches and automatically tracked the position, velocity, rotation axis, and spin of the ball along its trajectory. To demonstrate the validity of our system, we analyzed the consistency of our measurements with a physics-based model that predicts the trajectory of a spinning baseball. We found our measurements to coincide with the predicted positions to within an average error of less than a quarter of the baseball's diameter over the entire flight path. Accuracy is of high importance, since small differences in hand motion at launch time directly influence the ball's path. Due to its visualization component that shows an animation of the hand during ball release and of the resulting flight of the ball, the system is of value to athletes and coaches, allowing them to analyze and, subsequently, improve the athlete's performance.

Since most of our work is related to animation, videos naturally provide a better impression of the results than still images. Some movies documenting the described projects can be found at the following locations: `http://www.mpi-inf.mpg.de/resources/FAM/` features the MEDUSA facial animation system together with all research that originated from it. `http://www.mpi-inf.mpg.de/resources/VirtualHumans/` is dedicated to work concerned with tracking, modeling and animating virtual humans, including our hand model.

7.1 Future Challenges

As is usually the case in research, we set out to search for solutions, and did not only find answers, but also many interesting questions. A lot of exciting problems still wait to be explored. Below, we will sum up the main tasks for the issues addressed in this thesis.

7.1.1 Facial Animation

Most critique we got for our facial animations from lay persons concerned the eyes, or were complaints that movements were repetitive or robot-like. More realistic animations of the eyes

(for example, [LBB02]) and maybe a better eye model should alleviate the first problem, while more diverse animations would help with the second one. Here, it might also be beneficial to investigate differences between non-verbal speech-related facial expressions during spontaneous speech and during loud reading or recitation. As for the last point of critique, this seems to imply that the dynamics of the movements are too simplistic. Studying video recordings of speech accompanying facial expressions could lead to valuable insights and, for example, reveal several phases of movement with dynamical differences.
With respect to facial expressions of emotion, lifting the algorithm for the generation of intermediate facial expressions of emotion from Section 4.4.3 to the third dimension would be an interesting extension.
Emotion detection from the speech signal would allow to display emotions also in animations driven by a natural speech signal. Learning based approaches for a limited set of emotions were, for instance, proposed in [CDB02, CTFP05]. Extracting levels of activation, evaluation, and power instead of discrete emotions would allow more subtle facial expressions. Integrating our algorithm for intermediate facial expressions would guarantee integrity of animation and audio signal due to the underlying emotion model.
When facial expressions of emotion and speech are combined, tempo and frequency of non-verbal speech-related facial expressions should be adapted to the current emotion. Other parameters of emotion are, for example, blushing and frequency of breathing.

7.1.2 Hand Modeling and Animation

This area offers several important challenges, the most obvious being the necessity for texturing the hand models. Preferably, the method should permit to generate a texture for a personalized hand model from the same photograph and possibly the same feature points as the hand model itself, plus a second similarly tagged photograph of the palm.
Another issue is that of control. With so many degrees of freedom, guessing the correct muscle contraction values to obtain the desired animation can only be tedious. Starting from our model, Tsang et al. [TSF05] developed a biomechanical hand model where they solve for the differential equation of motion using the implicit Euler technique. They overcome the control problem by creating animations from keyframes using inverse dynamics. Compared to linear interpolation, using a physics-based anatomical model fills the gaps between the individual keyframes with animations that exhibit more realistic dynamics.
In this context, the problem of comparison to real data arises. One possible approach involves personalizing the hand model as described in Section 6.2.3. Movement of the same live hand is then tracked. Tsang et al. [TSF05] demonstrated that it is possible to find contraction values that yield an animation exhibiting dynamics very similar to the original movement. Results are good, but differences are clearly visible, probably due to measurement errors and (unavoidable) simplifications in the model.
This leads already to the next question: would a more complex model help? Would adding tendons, ligaments, connective tissue, skin layers or a more sophisticated muscle model improve the realism of the hand model or merely slow down animations? Depending on the intended application, trading interactivity for realism is certainly desirable, but whether a more complex model will also be more lifelike can only be decided by trying it out. In computer graphics, visual appearance is decisive, while for medical applications an exact rendition of the inner workings is of major importance.
The logical extension of this dissertation is the implementation of speech-related gestures,

thereby connecting our facial animations and our hand model, see Section 6.4.

7.2 Where Do We Go from Here?

The ultimate goal is a complete virtual human capable of human-like, trust evoking communication that puts the user at his ease by offering him the interface he knows best and is hence most comfortable with. A whole body model does not only permit facial expressions and hand gestures, but would allow the virtual human to transmit communicative signals also via the remaining channels of communication: through posture, personal space[1], timing[2], touching behavior, style of clothing, and artifacts such as jewelry. Both the scenario where a virtual character interacts with a human and where he deals with other agents have their own challenges. The former setting needs to integrate a good deal of computer vision, while the latter requires the virtual beings to be modeled as individuals, i.e. with different communication idiosyncrasies. In both cases, behavior must be adjusted to match the conversational partner. Non-verbal communication depends on the gender of the interlocutors, on their relative position in the hierarchy, on mood, personality, and other factors. As long as not all aspects are taken into account simultaneously, the mission is not completed.

[1] Distance kept between people during conversation; depends for instance on level of acquaintance and comfort.

[2] The meaning of time varies widely between different cultures; as a result, being on time, for example, is not of the same importance in all cultures, which may lead to misunderstanding.

Bibliography

[3D 03] 3D cafe. Arm skeleton. `http://www.3dcafe.com/asp/anatomy.asp`, 2003.

[ABM05] ABM United Kingdom Ltd. PROfit™. `http://www.abm-uk.com/uk/products/profit.asp`, 2005.

[AC76] M. Argyle and M. Cook. *Gaze and Mutual Gaze*. Cambridge University Press, Cambridge, England; New York, NY, 1976.

[AC99] J. Aggarwal and Q. Cai. Human motion analysis: a review. *Computer Vision and Image Unterstanding*, 73(3):428–440, 1999.

[Ada02] R. Adair. *The Physics of Baseball*. HarperCollins, New York, NY, 3rd edition, 2002.

[ADMH04] E. André, L. Dybkyaer, W. Minker, and P. Heisterkamp, editors. *Proc. Tutorial and Research Workshop on Affective Dialogue Systems 2004*, volume 3068 of *Lecture Notes in Artificial Intelligence*, Berlin and Heidelberg, Germany, 2004. Springer-Verlag.

[AHK87] J. Allen, S. Hunnicutt, and D. Klatt. *From text to speech: the MITalk system*. Cambridge University Press, Cambridge, England; New York, NY, 1987.

[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, 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. Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG) 2002*, pages 9–16, 2002.

[AHS03] I. Albrecht, J. Haber, and H.-P. Seidel. Construction and animation of anatomically based human hand models. In *Proc. Symp. Computer Animation 2003*, pages 98–109, 2003.

[Ala98] L. Alaways. *Aerodynamics of the Curve-Ball: An Investigation of the Effects of Angular Velocity on Baseball Trajectories*. PhD thesis, University of California, Davis, Davis, CA, 1998.

[AMH01] L. Alaways, S. Mish, and M. Hubbard. Identification of release conditions and aerodynamic forces in pitched-baseball trajectories. *J. Applied Biomechanics*, 17:63–76, 2001.

[And99] P. Andersen. *Nonverbal Communication*. Mayfield Publishing Company, Mountain View, CA, 1999.

[AS03] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. Computer Vision and Pattern Recognition 2003*, volume 2, pages 432–442, 2003.

[ASHS05] I. Albrecht, M. Schröder, J. Haber, and H.-P. Seidel. Mixed feelings: Expression of non-basic emotions in a muscle-based talking head. *J. Virtual Reality*, 8(4):201–212, 2005.

[Asp05] Aspley Ltd. E-FIT™. `http://www.efit.co.uk`, 2005.

[AT89] M. Allen and D. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, Oxford, United Kingdom, 1989.

[AUC+83] K. An, Y. Ueba, E. Chao, W. Cooney, and R. Linscheid. Tendon excursion and moment arm of index finger muscles. *J. Biomechanics*, 16(6):419–425, 1983.

[BA83] P. Burt and E. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graphics*, 2(4):217–236, 1983.

[BA92] B. Buchholz and T. Armstrong. A kinematic model of the human hand to evaluate its prehensile capabilities. *J. Biomechanics*, 25(2):149–162, 1992.

[BAHS] V. Blanz, I. Albrecht, J. Haber, and H.-P. Seidel. Creating face models from vague mental images. Under review.

[Bal81] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[Bar98] H. Baruh. *Analytical Dynamics*. McGraw-Hill, New York, NY, 1998.

[Bar01] D. Barr. Trouble in mind: paralinguistic indices of effort and uncertainty in communication. In *Oralité et Gestualité: Interactions et comportements multimodaux dans la communication. Actes du colloque ORAGE 2001*, pages 597–600, 2001.

[Bau87] E. Bauer. *Humanbiologie*. Cornelsen-Velhagen & Klasing, Berlin, Germany, 2nd edition, 1987.

[Bav94] J. Bavelas. Gestures as part of speech: methodological implications. *Research on Language and Social Interaction*, 27(3):201–221, 1994.

[BB02] M. Byun and N. Badler. FacEMOTE: qualitative parametric modifiers for facial animations. In *Proc. Symp. Computer Animation 2002*, pages 65–71, 2002.

[BBEO03] G. Bailly, M. Bérar, F. Elisei, and M. Odisio. Audiovisual speech synthesis. *Int'l J. Speech Technology*, 6(4):331–346, 2003.

[BBPV03] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Proc. Eurographics 2003)*, pages 641–650, 2003.

[BBT81] P. Brand, R. Beach, and D. Thompson. Relative tension and potential excursion of muscles in the forearm and hand. *J. Hand Surgery*, 6(3):209–219, 1981.

[BBW89] J. Burgoon, D. Buller, and W. Woodall. *Nonverbal Communication: The Unspoken Dialogue*. Harper & Row, New York, NY, 1989.

[BC95] A. Black and N. Campbell. Optimising selection of units from speech databases for concatenative synthesis. In *Proc. Eurospeech 1995*, volume 1, pages 581–584, 1995.

[BCS97] C. Bregler, M. Covell, and M. Slaney. Video Rewrite: driving visual speech with audio. In *Proc. SIGGRAPH 1997*, pages 353–360, 1997.

[BH99] P. Brand and A. Hollister. *Clinical Mechanics of the Hand*. Mosby, St. Louis, MO, 3rd edition, 1999.

[BHN04] T. Bui, D. Heylen, and A. Nijholt. Combination of facial movements on a 3D talking head. In *Proc. Computer Graphics International 2004*, pages 284–291, 2004.

[BHPN01] T. Bui, D. Heylen, M. Poel, and A. Nijholt. Generation of facial expressions from emotion using a fuzzy rule based system. In *Proc. Artificial Intelligence 2001*, pages 83–94, 2001.

[Bir71] R. Birdwhistell. *Kinesics and Context: Essays on Body-Motion Communication*. The Penguin Press, London, United Kingdom, 1971.

[BM96] R. Brunelli and O. Mich. SpotIt! An interactive identikit system. *Graphical Models and Image Processing*, 58(5):399–404, 1996.

[BNH+02] V. Bruce, H. Ness, P. Hancock, C. Newman, and J. Rarity. Four heads are better than one: combining face composites yields improvements in face likeness. *J. Applied Psychology*, 87(5):894–902, 2002.

[Boo97a] F. L. Bookstein. *Morphometric Tools for Landmark Data*. Cambridge University Press, Cambridge, England; New York, NY, 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. SIGGRAPH 1999*, pages 21–28, 1999.

[Bra00] T. Brants. TnT – a statistical part-of-speech tagger. In *Proc. Conf. Applied Natural Language Processing 2000*, pages 224–231, 2000.

[BS98a] E. Baker and M. Seltzer. The mug-shot search problem. In *Proc. Vision Interface 1998*, pages 65–72, 1998.

[BS98b] M. Brand and K. Shan. Voice-driven animation. In *Proc. Workshop on Perceptual User Interfaces*, 1998.

[BSVS04] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel. Exchanging faces in images. In *Proc. Eurographics 2004*, pages 669–676, 2004.

[BTC99] A. Black, P. Taylor, and R. Caley. Festival Speech Synthesis System, Edition 1.4. Technical report, Centre for Speech Technology Research, University of Edinburgh, United Kingdom, 1999.

[BV99] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH 1999*, pages 187–194, 1999.

[BV03] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[BY94] E. Biryukova and V. Yourovskaya. A model of hand dynamics. In F. Schuind, K. An, W. Cooney, and M. Garcia Elias, editors, *Advances in the Biomechanics of Hand and Wrist*, pages 107–122. Plenum Press, New York, NY, 1994.

[CB94] D. Collins and R. Bruce, editors. *Seeing the Unseen: Dr. Harold E. Edgerton and the Wonders of Strobe Alley*. MIT Press, Cambridge, MA, 1994.

[CBC+01] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. SIGGRAPH 2001*, pages 67–76, 2001.

[CC03] R. Cowie and R. Cornelius. Describing the emotional states that are expressed in speech. *Speech Communication. Special Issue on Speech and Emotion*, 40(1–2):5–32, 2003.

[CCZB00] D. Chi, M. Costa, L. Zhao, and N. Badler. The EMOTE model for effort and shape. In *Proc. SIGGRAPH 2000*, pages 173–182, 2000.

[CDB02] E. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proc. Pacific Graphics 2002*, pages 68–76, 2002.

[CDCA+99] R. Cowie, E. Douglas-Cowie, B. Appolloni, J. Taylor, A. Romano, and W. Fellenz. What a neural net needs to know about emotion words. In N. Mastorakis, editor, *Computational Intelligence and Applications*, pages 109–114. World Scientific and Engineering Society Press, 1999.

[CDCS+00] R. Cowie, E. Douglas-Cowie, S. Savvidou, E. McMahon, M. Sawey, and M. Schröder. 'FEELTRACE': an instrument for recording perceived emotion in real time. In *Proc. Int'l Speech Communication Assoc. Workshop on Speech and Emotion*, pages 19–24, 2000.

[CE05] Y.-J. Chang and T. Ezzat. Transferable videorealistic speech animation. In *Proc. Symp. Computer Animation 2005*, pages 143–151, 2005.

[CF04] T.-P. Chen and S. Fels. Exploring gradient-based face navigation interfaces. In *Proc. Graphics Interface 2004*, pages 65–72, 2004.

[CFKP04] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin. Real-time speech motion synthesis from recorded motions. In *Proc. Symp. Computer Animation 2004*, pages 347–355, 2004.

[CFP03] Y. Cao, P. Faloutsos, and F. Pighin. Unsupervised learning for speech motion editing. In *Proc. Symp. Computer Animation 2003*, pages 225–231, 2003.

[CG98] E. Cosatto and H. Graf. Sample-based synthesis of photo-realistic talking heads. In *Proc. Computer Animation 1998*, pages 103–110, 1998.

[CGB+96] C. Cavé, I. Guaïtella, R. Bertrand, S. Santi, F. Harlay, and R. Espesser. About the relationship between eyebrow movements and F0 variations. In *Proc. Int'l Conf. Spoken Language Processing 1996*, pages 2175–2179, 1996.

[Cha90] R. Chase. Examination of the hand and relevant anatomy. In *Plastic Surgery*, volume 7, chapter 89, pages 4247–4284. W. B. Saunders, Philadelphia, PA, 1990.

[Cho91] N. Chovil. Discourse-oriented facial displays in conversation. *Research on Language and Social Interaction*, 25:163–194, 1991.

[CJ91] C. Caldwell and V. Johnston. Tracking a criminal suspect through "face-space" with a genetic algorithm. In *Proc. Int'l Conf. Genetic Algorithms 1991*, pages 416–421, 1991.

[CLK01] B. Choe, H. Lee, and H.-S. Ko. Performance-driven muscle-based facial animation. *J. Visualization and Computer Animation*, 12:67–79, 2001.

[CM93] M. Cohen and D. Massaro. Modeling coarticulation in synthetic visual speech. In N. Magnenat-Thalmann and D. Thalmann, editors, *Models and Techniques in Computer Animation*, pages 139–156. Springer, Tokyo, 1993.

[CNB+01] J. Cassell, Y. Nakano, T. Bickmore, C. Sidner, and C. Rich. Non-verbal cues for discourse structure. In *Proc. Annual Meeting Assoc. Computational Linguistics 2001*, pages 106–115, 2001.

[Cos91] J. Cosnier. Les gestes de la question. In C. Kerbrat-Orecchioni, editor, *La Question*, pages 163–171. Presses Universitaires de Lyon, Lyon, France, 1991.

[CPB+94] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proc. SIGGRAPH 1994*, pages 413–420, 1994.

[CSB+94] J. Cassell, M. Steedman, N. Badler, C. Pelachaud, M. Stone, B. Douville, S. Prevost, and B. Achorn. Modeling the interaction between speech and gesture. In *Proc. Annual Conf. Cognitive Science Soc.*, 1994.

[CTFP05] Y. Cao, W. Tien, P. Faloutsos, and F. Pighin. Expressive speech-driven facial animation. *ACM Trans. Graphics*, 24(4):1283–1302, 2005.

[CVB01] J. Cassell, H. Vilhjálmsson, and T. Bickmore. BEAT: the Behavior Expression Animation Toolkit. In *Proc. SIGGRAPH 2001*, 2001.

[DACN02] T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences. In *Proc. Int'l Conf. Pattern Recognition 2002*, volume 1, pages 210–213, 2002.

[DC82] Graham Davies and Donald Christie. Face recall: an examination of some factors limiting composite production accuracy. *J. Applied Psychology*, 67(1):103–109, 1982.

[DCCCR03] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach. Emotional speech: towards a new generation of databases. *Speech Communication. Special Issue on Speech and Emotion*, 40(1–2):33–60, 2003.

[DCL81] K. Deffenbacher, T. Carr, and J. Leu. Memory for words, pictures, and faces: retroactive interference, forgetting, and reminiscence. *J. Experimental Psychology*, 7(4):299–305, 1981.

[DF77] S. Duncan and D. Fiske. *Face-to-Face Interaction*. Lawrence Earlbaum, Hillsdale, NJ, 1977.

[DiP02] S. DiPaola. FaceSpace: a facial spatial-domain toolkit. In *Proc. Information Visualisation 2002*, pages 105–109, 2002.

[DJVO00] K. Deffenbacher, J. Johanson, T. Vetter, and A. O'Toole. The face typicality-recognizability relationship: encoding or retrieval locus? *Memory and Cognition*, 28(7):1173–1182, 2000.

[Dor93] B. Dorner. Hand shape identification and tracking for sign language interpretation. In *Workshop on Looking at People – Int'l Joint Conf. Artificial Intelligence*, 1993.

[DPP+96] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. van der Vrecken. The MBROLA project: towards a set of high quality speech synthesisers free of use for non commercial purposes. In *Proc. Int'l Conf. Spoken Language Processing 1996*, pages 1393–1396, 1996.

[dR98] J.-P. de Ruiter. *Gesture and Speech Production*. PhD thesis, Katholieke Universiteit Nijmegen, Nijmegen, The Netherlands, 1998.

[dR00] J. de Ruiter. The production of gesture and speech. In D. McNeill, editor, *Language and Gesture: Window into Thought and Action*, pages 284–311. Cambridge University Press, Cambridge, England; New York, NY, 2000.

[DRSV02] D. DeCarlo, C. Revilla, M. Stone, and J. Venditti. Making discourse visible: coding and animating conversational facial displays. In *Proc. Symp. Computer Animation 2002*, pages 11–16, 2002.

[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.

[Dun74] S. Duncan. On the structure of speaker-auditor interaction during speaking turns. *Language in Society*, 3(2):161–180, 1974.

[Dut97] T. Dutoit. *An Introduction to Text-to-Speech Synthesis*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[EGP02] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. In *Proc. SIGGRAPH 2002*, pages 388–398, 2002.

[EK97] P. Ekman and D. Keltner. Universal facial expressions of emotion: an old controversy and new findings. In U. Segerstrôle and P. Molnár, editors, *Nonverbal Communication: Where Nature Meets Culture*, pages 27–46. Lawrence Erlbaum, Mahwah, NJ, 1997.

[Ekm65] P. Ekman. Differential communication of affect by head and body cues. *J. Personality and Social Psychology*, 2(5):726–735, 1965.

[Ekm79] P. Ekman. About brows: emotional and conversational signals. In M. v. Cranach, K. Foppa, W. Lepenies, and D. Ploog, editors, *Human Ethology: Claims and Limits of a New Discipline: Contributions to the Colloquium.*, pages 169–248. Cambridge University Press, Cambridge, England; New York, NY, 1979.

[ES03] G. ElKoura and K. Singh. Handrix: animating the human hand. In *Proc. Symp. Computer Animation 2003*, 2003.

[EW69] P. Ekman and W. Wallace. The repertoire of nonverbal behavior: categories, origins, usage, and coding. *Semiotica*, 1:49–98, 1969.

[Fau93] O. Faugeras. *Three-dimensional computer vision : a geometric viewpoint*. MIT Press, Cambridge, MA, 1993.

[FHC04] C. Frowd, P. Hancock, and D. Carson. EvoFIT: a holistic, evolutionary facial imaging technique for creating composites. *ACM Trans. Applied Perceptions*, 1(1):19–39, July 2004.

[Fit54] P. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *J. Experimental Psychology*, 47:381–391, 1954.

[Fun93] Y. Fung. *Biomechanics: Mechanical Properties of Living Tissues*. Springer-Verlag, New York, NY, 2nd edition, 1993.

[GC75] M. Gillenson and B. Chandrasekaran. A heuristic strategy for developing human facial images on a CRT. *Pattern Recognition*, 7(4):187–196, 1975.

[GFG+01] M. Gleicher, N. Ferrier, A. Gardner, S.Y. Shin, T. Tolles, and T. Wilson. Making motion capture useful. In *SIGGRAPH Course Notes*, 2001.

[GGW+98] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *Proc. Conf. Computer Graphics and Interactive Techniques 1998*, pages 55–66, 1998.

[Gla03] E. Gladilin. *Biomechanical Modeling of Soft Tissue and Facial Expressions for Craniofacial Surgery Planning*. PhD thesis, FU Berlin, Germany, 2003.

[GMTT89] J.-P. Gourret, N. Magnenat-Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. In *Proc. SIGGRAPH 1989*, pages 21–30, 1989.

[Gol02] H. Goldstein. *Classical Mechanics*. Prentice Hall, 3rd edition, 2002.

[GPBS03] S. Gibson, A. Pallares Bejarano, and C. Solomon. Synthesis of photographic quality facial composites using evolutionary algorithms. In *Proc. British Machine Vision Conf. 2003*, pages 221–230, 2003.

[GUAT98] F. Galanes, J. Unverferth, L. Arslan, and D. Talkin. Generation of lip-synched synthetic faces from phonetically clustered face movement data. In *Proc. Audio-Visual Speech Processing 1998*, pages 191–194, 1998.

[Gue02] A. Gueziec. Tracking pitches for broadcast television. *IEEE Computer*, 35(3):38–43, 2002.

[Gue03] A. Gueziec. Tracking a baseball for broadcast television. In *SIGGRAPH Course Notes*, 2003.

[Gus86] C. Gussenhoven. The intonation of 'George and Mildred': post-nuclear generalizations. In C. Johns-Lewis, editor, *Intonation in Discourse*, pages 77–123. College-Hill Press, San Diego, CA, 1986.

[GV93] C. Gerthsen and H. Vogel. *Physik*. Springer-Verlag, Berlin and Heidelberg, Germany, 13th edition, 1993.

[HBG01] D. House, J. Beskow, and B. Granström. Timing and interaction of visual cues for prominence in audiovisual speech perception. In *Proc. Eurospeech 2001*, pages 387–390, 2001.

[HBMTT95] Z. Huang, R. Boulic, N. Magnenat-Thalmann, and D. Thalmann. A multi-sensor approach for grasping and 3D interaction. In *Proc. Computer Graphics International 1995*, pages 235–254, 1995.

[HH86] T. Heap and D. Hogg. 3D deformable hand models. In *Proc. Gesture Workshop 1996*, pages 131–139, 1986.

[HMP02] B. Hartmann, M. Mancini, and C. Pelachaud. Formational parameters and adaptive prototype installation for MPEG-4 compliant gesture synthesis. In *Proc. Computer Animation 2002*, pages 111–119, 2002.

[HMP05] B. Hartman, M. Mancini, and C. Pelachaud. Implementing expressive gesture synthesis for embodied conversational agents. In *Proc. Gesture Workshop 2005*, pages 45–55, 2005.

[HNW93] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, New York, 2nd edition, 1993.

[Hor87] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Optical Soc. America*, 4(4), 1987.

[Hou00] T. House. *The Pitching Edge*. Human Kinetics, Champaign, IL, 2nd edition, 2000.

[HS96] J. Heikkila and O. Silven. Calibration procedure for short focal length off-the-shelf CCD cameras. In *Proc. Int'l Conf. Pattern Recognition 1996*, pages 166–170, 1996.

[HUM05] The HUMAINE network portal. http://emotion-research.net, 2005.

[IC96] H. Ip and C. Chan. Script-based facial gesture and speech animation using a NURBS based face model. *Computers and Graphics*, 20(6):881–891, 1996.

[ICL97] H. Ip, S. Chan, and M. Lam. HACS: Hand Action Coding System for anatomy-based synthesis of hand gestures. In *Proc. Int'l. Conf. Systems, Man, and Cybernetics 1998*, pages 1307–1312, 1997.

[ICL00] H. Ip, S. Chan, and M. Lam. Hand gesture animation from static postures using an anatomy-based model. In *Proc. Computer Graphics International 2000*, pages 29–36, 2000.

[IDE05] IDENTI.NET Internet + Software Services GmbH. FACETTE® Face Design System. http://www.facette.de/eng/index_html.html, 2005.

[Int02] Intel. Open Source Computer Vision Library. http://www.sourceforge.net/projects/opencvlibrary, 2002.

[IQ 05] IQ Biometrics. FACES. http://www.iqbiometrix.com, 2005.

[JD85] F. Jenkins and G. Davies. Contamination of facial memory through exposure to misleading composite pictures. *J. Applied Psychology*, 70(1):164–176, 1985.

[JKS95] R. Jain, R. Kasturi, and B. Schunck. *Machine Vision*. McGraw Hill International, New York, NY, 1995.

[JS99] T. Johnstone and K. Scherer. The effects of emotions on voice quality. In *Proc. Int'l Congress of Phonetic Sciences 1999*, pages 2029–2032, 1999.

[Käh03] K. Kähler. *A Head Model with Anatomical Structure for Facial Modeling and Animation*. PhD thesis, Universität des Saarlandes, Saarbrücken, 2003.

[KCC96] R. Krauss, Y. Chen, and P. Chawla. Nonverbal behavior and nonverbal communication: what do conversational hand gestures tell us? *Advances in Experimental Psychology*, 28:389–450, 1996.

[KCG00] R. Krauss, Y. Chen, and R. Gottesman. Lexical gestures and lexical access: a process model. In D. McNeill, editor, *Language and Gesture: Window into Thought and Action*, pages 261–283. Cambridge University Press, Cambridge, England; New York, NY, 2000.

[KCMT00] J. Kim, F. Cordier, and N. Magnenat-Thalmann. Neural network-based violonist's hand animation. In *Proc. Computer Graphics International 2000*, pages 37–41, 2000.

[Ken80] A. Kendon. Gesticulation and speech: two aspects of the process of utterance. In M. Key, editor, *The Relationship of Verbal and Nonverbal Communication*, pages 207–227. Mouton Publisher, The Hague, The Netherlands, 1980.

[Ken86] A. Kendon. Current issues in the study of gesture. In J.-L. Nespoulous, P. Perron, and A. Lecours, editors, *The Biological Foundations of Gestures: Motor and Semiotic Aspects*, pages 23–47. Lawrence Earlbaum, Hillsdale, NJ, 1986.

[Ken94] A. Kendon. Do gestures communicate? A review. *Research on Language and Social Interaction*, 27(3):175–200, 1994.

[KHS01] K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based muscle modeling for facial animation. In *Proc. Graphics Interface 2001*, pages 37–46, 2001.

[KHYS02] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: generating animated head models with anatomical structure. In *Proc. Symp. Computer Animation 2002*, pages 55–64, 2002.

[Kip03] M. Kipp. *Gesture Generation by Imitation: From Human Behavior to Computer Character Animation*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2003.

[KJP02] P. Kry, D. James, and D. Pai. EigenSkin: real time large deformation character skinning in hardware. In *Proc. Symp. Computer Animation 2002*, pages 153–159, 2002.

[KKM03] S. King, A. Knott, and B. McCane. Language-driven nonverbal communication in a bilingual conversational agent. In *Proc. Computer Animation and Social Agents 2003*, pages 17–22, 2003.

[KM04] T. Kurihara and N. Miyata. Modeling deformable human hands from medical images. In *Proc. Symp. Computer Animation 2004*, pages 357–365,406, 2004.

[KMMTT91] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. SMILE: a multilayered facial animation system. In *Proc. Int'l Federation for Information Processing Working Group 5.10*, pages 189–198, 1991.

[KMMTT92] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. In *Proc. Eurographics 1992*, pages 59–69, 1992.

[KMT00] S. Kshirsagar and N. Magnenat-Thalmann. Lip synchronization using linear predictive analysis. In *Proc. IEEE Int'l Conf. Multimedia and Expo*, volume 2, pages 1077–1080, 2000.

[KMvG04] G. Kalberer, P. Müller, and L. van Gool. Animation pipeline: realistic speech based on observed 3D face dynamics. In *Proc. European Conf. Visual Media Production*, pages 1–10, 2004.

[Koh92] K. Kohler. Gestural reorganization in connected speech: a functional viewpoint on 'articulatory phonology'. *Phonetica*, 49:205–211, 1992.

[KP04] B. Krenn and H. Pirker. Defining the Gesticon: language and gesture coordination for interacting embodies agents. In *Proc. Symp. Language, Speech and Gesture for Expressive Characters – Artificial Intelligence and Simulation of Behaviour 2004*, pages 107–115, 2004.

[KP05] S. King and R. Parent. Creating speech-synchronized animation. *The Visual Computer*, 11(3):341–352, 2005.

[KPG+02] B. Krenn, H. Pirker, M. Grice, P. Piwek, K. van Deemter, M. Schröder, M. Klesen, and E. Gstrein. Generation of multimodal dialogue for net environments. In *Proc. Konvens*, pages 91–98, 2002.

[KSV+01] E. Klabbers, K. Stöber, R. Veldhuis, P. Wagner, and S. Breuer. Speech synthesis development made easy: the Bonn Open Synthesis System. In *Proc. Eurospeech 2001*, volume 1, pages 521–524, 2001.

[KSW04] S. Kopp, T. Sowa, and I. Wachsmuth. Imitation games with an artificial agent: from mimicking to understanding shape-related iconic gestures. In *Proc. Gesture Workshop 2003*, volume 2915 of *LNCS*, pages 436–447, Berlin and Heidelberg, Germany, 2004. Springer.

[KTC04] S. Kopp, P. Tepper, and J. Cassell. Towards integrated microplanning of language and iconic gesture for multimodal output. In *Proc. Int'l Conf. Multimodal Interfaces 2004*, pages 97–104, 2004.

[KTM+93] T. Kunii, Y. Tsuchida, H. Matsuda, M. Shirahama, and S. Miura. A model of the hands and arms based on manifold mappings. In *Proc. Computer Graphics International 1993*, pages 381–398, 1993.

[KW00a] S. Kopp and I. Wachsmuth. A knowledge-based approach for lifelike gesture animation. In *Proc. European Conf. Artificial Intelligence 2000*, pages 663–667, 2000.

[KW00b] S. Kopp and I. Wachsmuth. Planning and motion control in lifelike gesture: a refined approach. In *Proc. Computer Animation 2000*, pages 92–97, 2000.

[KW02] S. Kopp and I. Wachsmuth. Model-based animation of coverbal gesture. In *Proc. Computer Animation 2002*, pages 252–257, 2002.

[KW04] S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents. *Computer Animation and Virtual Worlds*, 15:39–52, 2004.

[KYS03] S. Kettebekov, M. Yeasin, and R. Sharma. Improving continuous gesture recognition with spoken prosody. In *Proc. Computer Vision and Pattern Recognition 2003*, volume 1, pages 565–570, 2003.

[LAAB02] C. Latta, N. Alvaredo, S. Adams, and S. Burbeck. An expressive system for endowing robots or animated characters with affective facial displays. In *Proc. Animating Expressive Characters for Social Interaction – Artificial Intelligence and Simulation of Behaviour 2002*, 2002.

[Lan61] J. Landsmeer. Studies in the anatomy of articulation. *Acta morphologica Neerlando-Scandinavia*, 3:287–303, 1961.

[LB99] M. Lundeberg and J. Beskow. Developing a 3D-agent for the AUGUST dialogue system. In *Proc. Audio-Visual Speech Processing 1999*, 1999.

[LBB02] S. Lee, J. Badler, and N. Badler. Eyes alive. In *Proc. SIGGRAPH 2002*, pages 637–644, 2002.

[Lev89] W. Levelt. *Speaking*. MIT press, Cambridge, MA, 1989.

[LF80] Kenneth R. Laughery and Richard H. Fowler. Sketch artist and Identi-Kit procedures for recalling faces. *J. Applied Psychology*, 65(3):307–316, 1980.

[LG97] B. Le Goff. Automatic modeling of coarticulation in text-to-visual speech synthesis. In *Proc. Eurospeech 1997*, pages 1667–1670, 1997.

[LG99a] T. Lebourque and S. Gibet. A complete system for the specification and the generation of sign language gestures. In A. Braffort, R. Gherbi, S. Gibet, J. Richardson, and D. Teil, editors, *Gesture-Based Communication in Human-Computer Interaction*, volume 1739 of *LNAI*. Springer-Verlag, Berlin and Heidelberg, Germany, 1999.

[LG99b] T. Lebourque and S. Gibet. High level specification and control of communication gestures: the GESSYCA system. In *Proc. Computer Animation 1999*, pages 24–35, 1999.

[LK93] K.-H. Lee and K. Kroemer. A finger model with constant tendon moment arms. In *Proc. Human Factors and Ergonomics Soc. Annual Meeting 1993*, pages 710–714, 1993.

[Löf90] A. Löfqvist. Speech as audible gestures. In W. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modelling*, pages 289–322. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.

[LTW93] Y. Lee, D. Terzopoulos, and K. Waters. Constructing physics-based facial models of individuals. In *Proc. Graphics Interface 1993*, pages 1–8, 1993.

[LTW95] Y. Lee, D. Terzopoulos, and K. Waters. Realistic face modeling for animation. In *Proc. SIGGRAPH 1995*, pages 55–62, 1995.

[LWH00] J. Lin, Y. Wu, and T. Huang. Modeling the constraints of human hand motion. In *Proc. Workshop on Human Motion*, pages 121–126, 2000.

[McC94] E. McClave. Gestural beats: the rythm hypothesis. *J. Psycholinguistic Research*, 23(1), 1994.

[McC98] E. McClave. Pitch and manual gestures. *J. Psycholinguistic Research*, 27(1), 1998.

[MCM94] D. McNeill, J. Cassell, and K.-E. McCullough. Communicative effects of speech-mismatched gestures. *Research on Lanugage and Social Interaction*, 27(3):223–237, 1994.

[McN92] David McNeill. *Hand and Mind: What Gestures Reveal about Thought*. The University of Chicago Press, Chicago, 1992.

[McN02] D. McNeill. Gesture and language dialectic. *Acta Linguistica Hafniesia*, 2002.

[MFBLC01] Mulero Mártinez, J. Feliú Batlle, and J. López Coronado. Parametric neurocontroller for positioning of an antropomorfic finger based on an oponent driven-tendon transmission system. In *Proc. Int'l Work-Conf. Artificial Neural Networks 2001*, volume 1, pages 47–54, 2001.

[MLS94] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, 1994.

[MM76] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[MMT97] L. Moccozet and N. Magnenat-Thalmann. Dirichlet free-form deformations and their application to hand simulation. In *Proc. Computer Animation 1997*, pages 93–102, 1997.

[MSK92] P. Morrel-Samuels and R. Krauss. Word familiarity predicts temporal asynchrony of hand gestures and speech. *J. Experimental Psychology: Learning, Memory and Cognition*, 18:615–623, 1992.

[MTA+01] J. McDonald, J. Toro, K. Alkoby, A. Berthiaume, R. Carter, P. Chomwong, J. Christopher, M. Davidson, J. Furst, B. Konie, G. Lancaster, L. Roychoudhuri, E. Sedgewick, N. Tomuro, and R. Wolfe. An improved articulated model of the human hand. *The Visual Computer*, 17(3):158–166, 2001.

[MTLD88] N. Magnenat-Thalmann, R. Laperriére, and D.Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proc. Graphics Interface 1988*, pages 26–33, 1988.

[MTPT88] N. Magnenat-Thalmann, E. Primeau, and D. Thalmann. Abstract muscle action procedures for human face animation. *The Visual Computer*, 3(5):290–297, 1988.

[Muy87] E. Muybridge. *Animal Locomotion: An Electro-Photographic Investigation of Consecutive Phases of Animal Movements 1872–1885*. University of Pennsylvania, Philadelphia, PA, 1887.

[NEC05] Net Environment for Embodied Emotional Conversational Agents (NECA). `http://www.oefai.at/NECA/`, 2005.

[NN99] J. Noh and U. Neumann. A survey of facial modeling and animation techniques. USC Technical Report 99-705, University of Southern California, Los Angeles, CA, 1999.

[Nob98] S. Nobe. Synchrony between gestures and acoustic peaks of speech: a cross-linguistic study. In *Oralité et Gestualité: Communication multimodale, interaction. Actes du colloque ORAGE 1998*, pages 543–548, 1998.

[Nob00] S. Nobe. Where do *most* spontaneuos representational gestures actually occur with respect to speech? In D. McNeill, editor, *Language and Gesture: Window into Thought and Action*, pages 186–198. Cambridge University Press, Cambridge, England; New York, NY, 2000.

[NR04] H. Noot and Z. Ruttkay. Gesture in style. In *Proc. Gesture Workshop 2003*, volume 2915 of *LNAI*, pages 324–337, Berlin and Heidelberg, Germany, 2004. Springer-Verlag.

[NZB00] T. Noma, L. Zhao, and N. Badler. Design of a virtual human presenter. *IEEE Computer Graphics and Applications*, 20(4):79–85, 2000.

[OH98] H. Ouhaddi and P. Horain. Conception et ajustement d'un modèle 3D articulé de la main. In *Actes des 6èmes journées du Groupe de Travail Réalité Virtuelle*, volume 12/13, pages 83–90, 1998.

[OVBG97] D. Ostry, E. Vatikiotis-Bateson, and P. Gribble. An examination of the degrees of freedom of human jaw motion in speech and mastication. *J. Speech, Language, and Hearing Research*, 40:1341–1351, 1997.

[Par74] F. Parke. *A Parametric Model for Human Faces*. PhD thesis, University of Utah, Salt Lake City, UT, 1974.

[Par82] F. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, November 1982.

[PB81] S. Platt and N. Badler. Animating facial expressions. In *Proc. SIGGRAPH 1981*, pages 245–252, 1981.

[PBS91] C. Pelachaud, N. Badler, and M. Steedman. Linguistic issues in facial animation. In *Proc. Computer Animation 1991*, pages 15–30, 1991.

[PBS96] C. Pelachaud, N. Badler, and M. Steedman. Generating facial expressions for speech. *Cognitive Science*, 20(1):1–46, 1996.

[PF02] I. Pandzic and R. Forchheimer, editors. *MPEG-4 Facial Animation - The Standard, Implementations and Applications*. John Wiley & Sons, Hillsdale, NJ, USA, 2002.

[PHL+98] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proc. SIGGRAPH 1998*, pages 75–84, 1998.

[PJC98] G. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts. In *Proc. Computer Vision and Pattern Recognition 1998*, pages 260–265, 1998.

[PKS99] A. Paeschke, M. Kienast, and W. Sendlmeier. F0-contours in emotional speech. In *Proc. Int'l Congress of Phonetic Sciences 1999*, pages 929–931, 1999.

[Plu80] R. Plutchik. *Emotion: A Psychoevolutionary Synthesis*. Harper & Row, New York, NY, 1980.

[PLZ+89] S. Prillwitz, R. Leven, H. Zienert, T. Hamke, and J. Henning. HamNoSys Version 2.0: Hamburg Notation System for Sign Languages: an introductory guide. In *International Studies on Sign Language and Communication of the Deaf*, volume 5. Signum Press, Hamburg, Germany, 1989.

[PP00] I. Poggi and C. Pelachaud. Performative facial expressions in animated faces. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 155–188. MIT Press, Cambridge, MA, 2000.

[PP01] R. Putz and R. Pabst, editors. *Atlas of Human Anatomy — Volume 1: Head, Neck, Upper Limb*. Lippincott Williams & Wilkins, Philadelphia, PA, 13th edition, 2001.

[PP02] C. Pelachaud and I. Poggi. Subtleties of facial expressions in embodied agents. *J. Visualization and Computer Animation*, 13(5):301–312, 2002.

[PTVF92] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England; New York, NY, 2nd edition, 1992.

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

[PWHR98] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J.*, 16(5):295–306, 1998.

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

[PYOC00] G. Pingali, J. Yves, A. Opalach, and I. Carlbom. LucentVision: converting real world events into multimedia experiences. In *Proc. IEEE Int'l. Conf. Multimedia and Expo*, pages 1433–1436, 2000.

[PZ05] N. Pollard and V. Zordan. Physically based grasping control from example. In *Proc. Symp. Computer Animation 2005*, pages 311–318, 2005.

[QMB+02] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K. McCullough, and R. Ansari. Multimodal human discourse: gesture and speech. *ACM Trans. Computer-Human Interaction*, 9(3):171–193, 2002.

[RG91] H. Rijpkema and M. Girard. Computer animation of knowledge-based human grasping. In *Proc. SIGGRAPH 1991*, pages 339–348, 1991.

[RK94] J. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *Proc. European Conf. Computer Vision 1994*, volume 2, pages 35–46, 1994.

[RNtH03] Z. Ruttkay, H. Noot, and P. ten Hagen. Emotion Disc and Emotion Squares: tools to explore the facial expression space. *Computer Graphics Forum*, 22(1):49–53, 2003.

[RS91] B. Rimé and L. Schiaratura. Gesture and speech. In R. Feldman and B. Rimé, editors, *Fundamentals of Nonverbal Behavior*, pages 239–281. Cambridge University Press, Cambridge, England; New York, NY, 1991.

[SB98] W. Skut and T. Brants. Chunk tagger – statistical recognition of noun phrases. In *Proc. ESSLLI Workshop on Automated Acquisition of Syntax and Parsing*, 1998.

[SCDC+01] M. Schröder, R. Cowie, E. Douglas-Cowie, M. Westerdijk, and S. Gielen. Acoustic correlates of emotion dimensions in view of speech synthesis. In *Proc. Eurospeech 2001*, volume 1, pages 87–90, 2001.

[Sch00] K. Scherer. Psychological models of emotion. In J. Borod, editor, *The Neuropsychology of Emotion*, pages 137–162. Oxford University Press, New York, NY, 2000.

[Sch01] M. Schröder. Emotional speech synthesis: a review. In *Proc. Eurospeech 2001*, volume 1, pages 561–564, 2001.

[Sch04a] M. Schröder. Dimensional emotion representation as a basis for speech synthesis with non-extreme emotions. In *Proc. Workshop on Affective Dialogue Systems*, pages 209–220, 2004.

[Sch04b] M. Schröder. *Speech and emotion research: an overview of research frameworks and a dimensional approach to emotional speech synthesis*. PhD thesis, Universiät des Saarlandes, Saarbrücken, Germany, 2004.

[Sch05] M. Schröder. The MARY Text-to-Speech System. `http://mary.dfki.de`, 2005.

[SDO+04] M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. Speaking with hands: creating animated conversational characters from recordings of human performance. In *Proc. SIGGRAPH 2004*, pages 506–513, 2004.

[Sjö01] K. Sjölander. The Snack Sound Toolkit. `http://www.speech.kth.se/snack/`, 1997–2001.

[Sla70] J. Slansky. Recognition of convex blobs. *Pattern Recognition*, 2:3–10, 1970.

[SMC01] B. Stenger, P. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *Proc. Computer Vision and Pattern Recognition 2001*, volume 2, pages 310–315, 2001.

[Smi05] Smith & Wesson®. Identi-Kit.NET™. `https://www.identikit.net/jsp/Public/dsp_IdentiKit.jsp?Action=Info`, 2005.

[SNF05] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. In *Proc. SIGGRAPH 2005*, pages 417–425, 2005.

[SPCM97] F. Scheepers, R. Parent, W. Carlson, and S. May. Anatomy-based modeling of the human musculature. In *Proc. SIGGRAPH 1997*, pages 163–172, 1997.

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

[SSZ99] T. Schiebler, W. Schmidt, and K. Zilles. *Anatomie*. Springer-Verlag, Berlin, Germany, 8th edition, 1999.

[ST94] R. Mas Sanso and D. Thalmann. A hand control and automatic grasping system for synthetic actors. *Computer Graphics Forum*, 13(3):167–177, 1994.

[ST03] M. Schröder and J. Trouvain. The German text-to-speech synthesis system MARY: a tool for research, development and teaching. *Int'l J. Speech Technology*, 6:365–377, 2003.

[Sta00] J. Stallo. Simulating emotional speech for a talking head. Honour's Thesis, School of Computing, Curtin University of Technology, Bentley, Australia, 2000.

[Ste02] J. Stewart. *The Pitching Clinic*. Burford Books, Short Hills, NJ, 2002.

[Str94] J. Streeck. Gesture as communication II: The audience as co-author. *Research on Language and Social Interaction*, 27(3):201–221, 1994.

[STSL02] L. Sibille, M. Teschner, S. Srivastava, and J.-C. Latombe. Interactive simulation of the human hand. In *Proc. Computer Assisted Radiology and Surgery 2002*, pages 7–12, 2002.

[SWVG02] M. Simmons, J. Wilhelms, and A. Van Gelder. Model-based reconstruction for creature animation. In *Proc. Symposion on Computer Animation 2002*, pages 139–146, 2002.

[TAH+05] C. Theobalt, I. Albrecht, J. Haber, M. Magnor, and H.-P. Seidel. Pitching a baseball - tracking high-speed motion with multi-exposure images. In *Proc. SIGGRAPH 2004*, pages 538–547, 2005.

[TBM+88] D. Thompson, W. Buford, L. Myers, D. Giurintano, and J. Brewer III. A hand biomechanics workstation. In *Proc. SIGGRAPH 1988*, pages 335–343, 1988.

[Tho81] D. Thompson. Biomechanics of the Hand. *Perspectives in Computing*, 1(3):12–19, 1981.

[TRK+02] N. Tsapatsoulis, A. Raousaiou, S. Kollias, R. Cowie, and E. Douglas-Cowie. Emotion recognition and synthesis based on MPEG-4 FAPs. In I. Pandzic and R. Forchheimer, editors, *MPEG-4 Facial Animation - The Standard, Implementations and Applications*, pages 141–167. John Wiley & Sons, Hillsdale, NJ, USA, 2002.

[TSF05] W. Tsang, K. Singh, and E. Fiume. Helping Hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Proc. Symp. Computer Animation 2005*, pages 110–119, 2005.

[TW90] D. Terzopoulos and K. Waters. Physically-based facial modelling, analysis, and animation. *J. Visualization and Computer Animation*, 1(2):73–80, 1990.

[UNI05] UNIDAS. PHANTOM PROFESSIONALxp©. `http://www.unidas.com/html/phantome.html`, 2005.

[VBPP05] D. Vlasic, M. Brand, H. Pfister, and J. Popvić. Face transfer with multilinear models. In *Proc. SIGGRAPH 2005*, pages 426–433, 2005.

[Wag88] C. Wagner. The pianist's hand: anthropometry and biomechanics. *Ergonomics*, 31(1):97–131, 1988.

[WAL+94] J. Wu, Y. Ang, P. Lam, H. Loh, and A. Narasimhalu. Inference and retrieval of facial images. *Multimedia Systems*, 2:1–14, 1994.

[Wal98] H. Wallbott. Bodily expression of emotion. *European J. Social Psychology*, 28:879–896, 1998.

[Wat87] K. Waters. A muscle model for animating three-dimensional facial expression. In *Proc. SIGGRAPH 1987*, pages 17–24, 1987.

[WF95] K. Waters and J. Frisbie. A coordinated muscle model for speech animation. In *Proc. Graphics Interface 1995*, pages 163–170, 1995.

[WH01] Y. Wu and T. Huang. Hand modeling, analysis, and recognition. *IEEE Signal Processing Magazine*, 18(3):51–60, 2001.

[Whi89] C. Whissell. The dictionary of affect in language. In R. Plutchik and H. Kellerman, editors, *Emotion: Theory, Research, and Experience*, volume 4: The Measurement of Emotions, chapter 5, pages 113–131. Academic Press, San Diego, CA, 1989.

[WHL+04] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-D facial expressions. In *Proc. Eurographics 2004*, pages 677–686, 2004.

[Wil90] L. Williams. Performance-driven facial animation. In *Proc. SIGGRAPH 1990*, pages 235–242, 1990.

[WL93] K. Waters and T. Levergood. DECface: An Automatic Lip-Synchronization Algorithm for Synthetic Faces. Technical Report 93-4, Cambridge Research Laboratories, Cambridge, MA, 1993.

[WLH01] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *Proc. Int'l Conf. Computer Vision 2001*, pages 426–432, 2001.

[WV97] J. Wilhelms and A. Van Gelder. Anatomically Based Modeling. In *Proc. SIGGRAPH 1997*, pages 173–180, 1997.

[Wyn01] C. Wynn. Implementing Bump-Mapping using Register Combiners. `http://www.nvidia.com/developer/`, 2001.

A

Pseudo Muscles of the Hand Model

anatomical name	ℓ_0 **[mm]**	**joint(s) / DOF**	$\|\|\vec{r}\|\|$ **[mm]**
flexor carpi radialis	52	wrist flexion	17.5
		wrist abduction	10.5
palmaris longus	50	wrist flexion	21
		wrist abduction	1.5
flexor digitorum superficialis index	72	wrist flexion	15
		wrist adduction	3
		MCP index flexion	11.9
		MCP index adduction	3
		PIP index flexion	6.2
flexor digitorum superficialis middle	70	wrist flexion	15
(analogous:		wrist adduction	3.0
flexor digitorum superficialis ring		MCP middle flexion	11.9
flexor digitorum superficialis pinky)		MCP middle adduction	1.7
		PIP middle flexion	6.2
flexor carpi ulnaris	42	wrist flexion	18.5
		wrist adduction	15
flexor digitorum profundus index	66	wrist flexion	6
(analogous:		wrist adduction	13
flexor digitorum profundus middle		MCP index flexion	11.1
flexor digitorum profundus ring		MCP index adduction	6
flexor digitorum profundus pinky)		PIP index flexion	7.9
		DIP index flexion	4.1
flexor pollicis longus	59	wrist flexion	5
		wrist abduction	13
		CMC thumb adduction	10
		CMC thumb opposition	10
		MCP thumb flexion	7.5
		IP thumb flexion	5.5
extensor carpi radialis longus	93	wrist extension	10
		wrist abduction	21
extensor carpi radialis brevis	61	wrist extension	13
		wrist abduction	24
extensor digitorum index	55	wrist extension	13
(analogous:		wrist adduction	7.5

Table A.1: **Pseudo muscle parameters.** List of the pseudo muscles of our system with fiber resting lengths ℓ_0 [mm], affected joints, and moment arms $||\vec{r}||$ [mm]. Source: [BH99, AUC$^+$83].

anatomical name	ℓ_0 **[mm]**	**joint(s) / DOF**	$\|\|\vec{r}\|\|$ **[mm]**
extensor digitorum middle		MCP index extension	8.6
extensor digitorum ring		MCP index abduction	0.2
extensor digitorum pinky)		PIP index extension	2.8
		DIP index extension	2.2
extensor digiti minimi	59	wrist extension	13
		wrist adduction	7.5
		MCP pinky extension	8.6
		PIP pinky extension	2.6
		DIP pinky extension	1.9
extensor carpi ulnaris	45	wrist extension	6
		wrist adduction	25
extensor pollicis longus	57	wrist extension	9
		wrist abduction	10.5
		CMC thumb extension	5
		CMC thumb adduction	10
		MCP thumb extension	2.5
		IP thumb extension	2
extensor indicis	55	wrist flexion	1.4
		wrist abduction	0.4
		MCP index extension	9
		MCP index adduction	1.3
		PIP index extension	2.6
		DIP index extension	1.9
abductor pollicis longus	46	wrist flexion	7.4
		wrist abduction	24
		CMC thumb extension	0.5
extensor pollicis brevis	43	wrist flexion	3.2
		wrist abduction	23
		CMC thumb extension	4.5
		CMC thumb abduction	3
		MCP thumb extension	3
abductor digiti minimi	40	CMC pinky opposition	6
		MCP pinky abduction	4
		PIP pinky extension	2.5
		DIP pinky extension	2
flexor digiti minimi brevis	34	CMC pinky opposition	6
		MCP index flexion	4
		MCP index abduction	4
opponens digiti minimi	34	CMC pinky opposition	6
abductor pollicis brevis	37	CMC thumb opposition	3.5
		CMC thumb abduction	7.5
		MCP thumb flexion	1
flexor pollicis brevis	36	CMC thumb opposition	9
		CMC thumb adduction	1
		MCP thumb flexion	7
opponens pollicis	24	CMC thumb opposition	4
		CMC thumb adduction	8.5

Table A.1: **Pseudo muscle parameters, continued.**

anatomical name	ℓ_0 **[mm]**	**joint(s) / DOF**	$\|\|\vec{\mathbf{r}}\|\|$ **[mm]**
adductor pollicis	36	CMC thumb opposition	4.5
		CMC thumb adduction	9
		MCP thumb flexion	7
lumbrical I	55	MCP index flexion	9.3
		MCP index radial abduction	4.8
		PIP index extension	1.8
		DIP index extension	0.7
lumbrical II	66	MCP middle flexion	5
		MCP middle radial abduction	4.8
		PIP middle extension	1.8
		DIP middle extension	0.7
lumbrical III	60	MCP ring flexion	5
		MCP ring radial abduction	4.8
		PIP ring extension	1.8
		DIP ring extension	0.7
lumbrical IV	49	MCP pinky flexion	5
		MCP pinky radial abduction	4.8
		PIP pinky extension	1.8
		DIP pinky extension	0.7
palmar interosseus I	15	MCP index flexion	6.6
		MCP index adduction	5.8
		DIP index extension	2.6
		PIP index extension	1.6
palmar interosseus II	15	MCP ring flexion	6.6
		MCP ring adduction	5.8
		DIP ring extension	2.6
		PIP ring extension	1.6
palmar interosseus III	15	MCP pinky flexion	6.6
		MCP pinky adduction	5.8
		DIP pinky extension	2.6
		PIP pinky extension	1.6
dorsal interosseus I	25	MCP index flexion	3.7
		MCP index abduction	6.1
		PIP index extension	2.6
		DIP index extension	1.6
dorsal interosseus II	25	MCP middle flexion	3.7
		MCP middle radial adduction	6.1
		PIP middle extension	2.6
		DIP middle extension	1.6
dorsal interosseus III	25	MCP middle flexion	3.7
		MCP middle ulnar adduction	6.1
		PIP middle extension	2.6
		DIP middle extension	1.6
dorsal interosseus VI	25	MCP pinky flexion	3.7
		MCP pinky abduction	6.1
		PIP pinky extension	2.6
		DIP pinky extension	1.6

Table A.1: **Pseudo muscle parameters, continued.**

B

Publications

The work presented in this thesis was published in the following papers.

[1] Volker Blanz, Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. *Creating Face Models from Vague Mental Images.* Under review.

[2] Irene Albrecht, Marc Schröder, Jörg Haber, and Hans-Peter Seidel. *Mixed feelings: Expression of non-basic emotions in a muscle-based talking head.* Journal of Virtual Reality 8(4). Special Issue on Language, Speech and Gesture for Virtual Reality, pages 201–212, 2005.

[3] Christian Theobalt, Irene Albrecht, Jörg Haber, Marcus Magnor and Hans-Peter Seidel. *Pitching a Baseball - Tracking High-Speed Motion with Multi-Exposure Images.* In: Proc. SIGGRAPH 2004, 2004, pages 538–547.

[4] Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. *Construction and Animation of Anatomically Based Human Hand Models.* In: Proc. Symp. Computer Animation 2003, 2003, pages 98–109,368.

[5] Irene Albrecht, Jörg Haber, Kolja Kähler, Marc Schröder and Hans-Peter Seidel. *"May I talk to you? :-)" – Facial Animation from Text.* In: Proc. Pacific Graphics 2002, 2002, pages 77–86.

[6] Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. *Automatic Generation of Non-Verbal Facial Expressions from Speech.* In: Proc. Computer Graphics International 2002, 2002, pages 283–293.

[7] Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. *Speech Synchronization for Physics-based Facial Animation.* In: Proc. 10th Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2002), 2002, pages 9–16.

[8] Jörg Haber, Kolja Kähler, Irene Albrecht, Hitoshi Yamauchi and Hans-Peter Seidel. *Face to Face: From Real Humans to Realistic Facial Animation.* In: Proc. 3rd Israel-Korea Binat'l Conf. on Geometrical Modeling and Computer Graphics, 2001, pages 73–82.

In addition, the facial composite system from Chapter 5 was presented as a sketch at SIGGRAPH 2005:

Irene Albrecht, Volker Blanz, Jörg Haber, and Hans-Peter Seidel. *Creating Face Models from Vague Mental Images.* SIGGRAPH sketch, 2005.