

Diss. ETH No. 19327

Measurement-Based Modeling and Fabrication of Deformable Materials for Human Faces

A dissertation submitted to
ETH Zurich

for the Degree of
Doctor of Sciences

presented by
Bernd Bickel
Dipl.Inf.Ing. ETH Zurich, Switzerland
born 03 June 1982
citizen of Austria

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Prof. Dr. Hanspeter Pfister, co-examiner
Dr. Wojciech Matusik, co-examiner

2010

Abstract

This thesis investigates the combination of data-driven and physically based techniques for acquiring, modeling, and animating deformable materials, with a special focus on human faces. Furthermore, based on these techniques, we introduce a data-driven process for designing and fabricating materials with desired deformation behavior.

Realistic simulation behavior, surface details, and appearance are still demanding tasks. Neither pure data-driven, pure procedural, nor pure physical methods are best suited for accurate synthesis of facial motion and details (both for appearance and geometry), due to the difficulties in model design, parameter estimation, and desired controllability for animators. Capturing of a small but representative amount of real data, and then synthesizing diverse on-demand examples with physically-based models and real data as input benefits from both sides: Highly realistic model behavior due to real-world data and controllability due to physically-based models.

To model the face and its behavior, hybrid physically-based and data-driven approaches are elaborated. We investigate surface-based representations as well as a solid representation based on FEM. To achieve realistic behavior, we propose to build light-weighted data capture devices to acquire real-world data to estimate model parameters and to employ concepts from data-driven modeling techniques and machine learning. The resulting models support

simple acquisition systems, offer techniques to process and extract model parameters from real-world data, provide a compact representation of the facial geometry and its motion, and allow intuitive editing. We demonstrate applications such as capture of facial geometry and motion and real-time animation and transfer of facial details, and show that our soft tissue model can react to external forces and produce realistic deformations beyond facial expressions.

Based on this model, we furthermore introduce a data-driven process for designing and fabricating materials with desired deformation behavior. The process starts with measuring deformation properties of base materials. Each material is represented as a non-linear stress-strain relationship in a finite-element model. For material design and fabrication, we introduce an optimization process that finds the best combination of base materials that meets a user's criteria specified by example deformations. Our algorithm employs a number of strategies to prune poor solutions from the combinatorial search space. We finally demonstrate the complete process by designing and fabricating objects with complex heterogeneous materials using modern multi-material 3D printers.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Kombination von datengetriebenen und physikalisch basierten Methoden, um deformierbare Materialien, insbesondere für die Beschreibung von menschlichen Gesichtern, zu erfassen und deren Deformationseigenschaften zu messen, zu modellieren und zu animieren. Weiters wird, gestützt auf diese Methoden, ein auf Messungen beruhender Prozess vorgestellt, um Materialien mit gewünschtem Deformationsverhalten zu designen und mittels Rapid-Prototyping-Verfahren herzustellen.

Weder rein datengetriebene, noch rein prozedurale oder rein physikalische Methoden sind optimal geeignet für die exakte Simulation von Gesichtsbewegungen und von hochaufgelösten Gesichtsdetails. Bestehende Modelle sind entweder schwierig zu implementieren und sehr rechenaufwendig, Modellparameter können nur sehr schwer geschätzt werden, oder entsprechende Methoden bieten wenig Kontroll- und Editiermöglichkeiten. Durch die Kombination einer kleinen, aber repräsentativen Menge an realen Daten und physikalisch basierten Simulationsmethoden kann jedoch von beiden Seiten profitiert werden. Somit wird realistisches und hochaufgelöstes Simulationsverhalten erreicht, Modellparameter können einfach und schnell erfasst und Editiermöglichkeiten bereitgestellt werden.

Daher werden hybride Ansätze erarbeitet, die auf physikalischen und datengetriebenen Grundlagen beruhen. Es werden sowohl auf Oberflächen basierende Repräsentationen als auch volumetrische Repräsentationen, letztere in Verbindung mit einer Finite-Elemente-Methode, untersucht. Zudem werden einfache Messsysteme präsentiert, um, gestützt auf realen Daten, Modellparameter zu schätzen und Konzepte aus dem Bereich datengetriebene Modellierung und maschinelles Lernen anzuwenden. Die daraus resultierenden Modelle bieten eine kompakte Repräsentation der Gesichtsgeometrie, deren Bewegungen und Deformationseigenschaften.

Als mögliche Anwendungsszenarien werden insbesondere die Erfassung von Geometrie und Bewegung menschlicher Gesichter, Echtzeitanimation und die Übertragung von Gesichtsdetails demonstriert. Weiters wird gezeigt, dass das vorgeschlagene Modell für Weichgewebe auch auf externe Krafteinflüsse reagieren kann und realistische Deformationen jenseits gewöhnlicher Gesichtsausdrücke synthetisiert werden können.

Auf diesem Modell aufbauend wird zudem ein datengetriebener Prozess für das Design und die Herstellung von Materialien mit gewünschtem Deformationsverhalten eingeführt. Der Prozess startet mit der Messung von Deformationseigenschaften einer diskreten Menge an Basismaterialien. Jedes Material wird als ein nichtlinearer Zusammenhang von Spannung und Dehnung repräsentiert. Für das Design von Materialien wird ein Optimierungsprozess vorgestellt, der versucht, eine optimale Kombination von Basismaterialien zu finden, um vorgegebene Beispieldeformationen zu approximieren. Dabei werden mehrere Strategien angewandt, um den kombinatorischen Raum an möglichen Lösungen zu beschränken. Der komplette Prozess wird anhand mehrerer Beispiele demonstriert, indem zuerst die Materialeigenschaften realer Objekte gemessen werden. Diese werden mittels des vorgeschlagenen Designprozesses approximiert, dann unter Verwendung eines modernen Multimaterial 3D Drucksystems für Rapid Prototyping hergestellt und schlussendlich mit Hilfe der vorgestellten Messsysteme validiert.

Acknowledgements

First of all, I would like to sincerely thank my advisor Prof. Markus Gross. His lectures sparked my interest in computer graphics, his apparent never ending interest in open research problems was very inspiring, and his unconditional support and guidance was invaluable during my Ph.D.

Furthermore, I would like to thank my close collaborators Prof. Hanspeter Pfister, Prof. Wojciech Matusik, Prof. Miguel A. Otaduy, and Prof. Mario Botsch. I really enjoyed working and discussing research with you! Their support, advice, and motivation was the fundamental foundation of my thesis.

It was also a pleasure to work with Roland Angst, Manuel Lang, and Moritz Bächer. It was a unique experience discussing and coding with them, and I want to thank them for their hard work and unconditional commitment to the projects we did together, especially when it came close to the deadline.

I owe all people at the Computer Graphics Laboratory and Disney Research Zurich a dept of gratitude. Thanks to Jeroen van Baar, Gian-Marco Baschera, Paul Beardsley, Thabo Beeler, Manuela Cavegn, Daniel Cotting, Marcel Germann, Simon Heinzle, Alexander Hornung, Wojciech Jarosz, Silke Konsorski-Lang, Doo Young Kwon, Gioacchino Noris, Thomas Oskam, Cengiz Oztireli, Tobias Pfaff, Steven Poulakos, Filip Sadlo, Johannes Schmid, Denis Steine-mann, Bob Sumner, Michael Waschbüsch, and Remo Ziegler.

A very special thanks to my office mates Martin Wicke and Sebastian Martin. We had the party office, and I enjoyed all our discussions about ongoing research projects! Very special thanks also to Peter Kaufmann for allowing me to use his development framework.

Furthermore, I would like to thank Janet McAndless for scanning parts of our subjects, Basil Weber for extending PBRT to support the skin-reflectance model, and our patient actors.

Finally, I want to mention my family and my girlfriend Martina for all their support and understanding.

This research has been supported by the NCCR Co-Me of the Swiss National Science Foundation.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Contributions	4
1.3	Organization	6
1.4	Publications	9
2	Fundamentals and Related Work	11
2.1	Capture	11
2.2	Modeling for Animation	14
2.3	Modeling Soft Tissue	16
2.4	Material Design and Fabrication	18
3	Modeling Facial Geometry and Motion	21
3.1	Motivation	22
3.2	Large-Scale Animation	23
	3.2.1 Linear Deformation Model	24
3.3	Medium-Scale Animation	27
	3.3.1 3D Wrinkle Synthesis	27
	3.3.2 Wrinkle Removal	29
3.4	Discussion	30

Contents

4	Performance Capture	33
4.1	Overview	33
4.2	Data Acquisition	34
4.2.1	Mocap / Face Correspondence	36
4.3	Medium-Scale Feature Reconstruction	38
4.3.1	Wrinkle Tracking	38
4.3.2	Cross-Section Shape Estimation	40
4.4	Results	42
4.4.1	Performance	42
4.4.2	Expression Replay and Wrinkle Editing	42
4.5	Discussion	43
5	Real-time Animation and Transfer of Facial Details	47
5.1	Motivation	48
5.2	Overview	49
5.3	Large-Scale Deformation	51
5.4	Fine-Scale Deformation in Pose Space	52
5.4.1	Definition of the Feature Vector	53
5.4.2	Pose-Space Deformation	54
5.4.3	Weighted Pose-Space Deformation	55
5.4.4	Transfer of Fine-Scale Details	56
5.5	Results and Implementation	57
5.5.1	Evaluation and Comparisons	57
5.5.2	Performance	59
5.5.3	Application Experiments	61
5.6	Discussion	63
6	Measuring Deformation Behavior	67
6.1	Requirements	67
6.2	Hand-held Device	68
6.2.1	Trinocular Stereo Vision System	69
6.2.2	Processing Pipeline	70
6.3	Automated Setup	71
6.3.1	Processing Pipeline	72
6.4	Discussion	72
7	Modeling of Non-Linear Heterogeneous Soft Tissue	75
7.1	Introduction	76
7.2	Modeling of Non-Linear Materials	77
7.2.1	Overview of our Approach	78
7.2.2	Discretization and Parameterization	79
7.2.3	Strain-Space Interpolation	81

7.2.4	Elastostatic FEM Simulation	82
7.3	Fitting the Material Parameters	83
7.3.1	Parameter Estimation Algorithm	83
7.3.2	Strain-Space Basis Selection	84
7.4	Results	85
7.4.1	Model Evaluation	85
7.4.2	Facial Deformation	87
7.4.3	Test for Path-Independence	87
7.5	Discussion	92
8	Design and Fabrication of Materials with Desired Deformation Behavior	95
8.1	Introduction	96
8.2	Non-Linear Material Model	98
8.3	Fitting Base Materials	100
8.4	Goal-Based Material Design	102
8.4.1	Formulation of the Problem	102
8.4.2	Branch-and-Bound with Clustering	102
8.5	Base Materials	105
8.6	Validation and Results	108
8.7	Discussion	112
9	Conclusion	119
9.1	Discussion	119
9.2	Future Work	121
A	Notation And Glossary	123
A.1	Operators	123
A.2	Notation	123
A.3	Glossary	126
B	Jacobian for Parameter Fitting	127
C	Curriculum Vitae	129
	List of Figures	133
	List of Tables	137
	Bibliography	139

Introduction

The human face is one of the biggest challenges in computer graphics. Since the early work of Parke [Parke, 1974], there has been a significant effort for capturing, modeling, and simulating its deformation behavior for a wide range of applications, from rendering photo realistic images and creating convincing animations, to surgery planning and computing control signals for expressive robotic heads.

The complexity of reproducing a believable human face is due to perceptual issues and the inherent complexity of skin deformations and the underlying facial tissue. *Perceptual studies* show that humans are especially good at detecting and recognizing subtle facial expressions. A twitch of an eye or a glimpse of a smile are subtle but important aspects of human communication and might occur in a fraction of a second. Both the *dynamics* of the expression and the *detailed spatial deformations* convey personality and intensity [Essa and Pentland, 1997]. As illustrated in Figure 1.1, synthetic human faces missing those subtle features are affected by an effect called the uncanny valley, causing negative emotional responses of the observer. Crossing the uncanny valley requires mastering a wide range of aspects, including 3D surface geometry, appearance, motion, and underlying physics, as illustrated in Figure 1.2. Each of those aspects encompasses a variety of research challenges, actively pursued by numerous research groups in computer graphics,

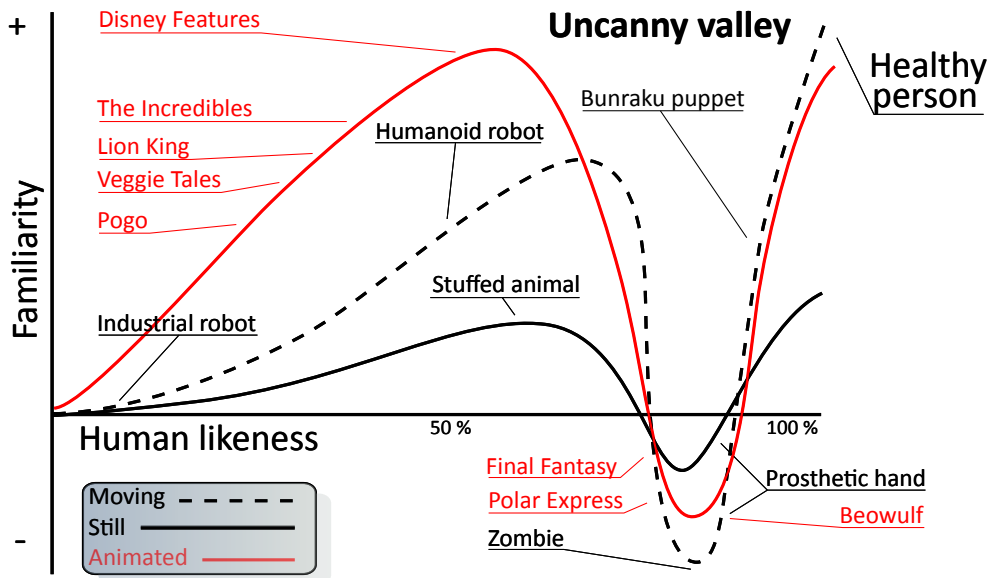


Figure 1.1: Uncanny valley. The graph emerged from the field of robotics [Mori, 1970] and visualizes the hypothesis that robots looking and moving almost but not entirely like humans cause a negative emotional response. The chart is augmented with a curve of animated films, where similar effects can be observed [Joly, 2010].

computer vision, and related fields such as biomechanics and perceptual research. Modeling those accurately is challenging because *biomechanical and anatomical studies* show that the face consists of several different highly heterogeneous tissue types, with anisotropic non-linear deformation behavior and complex dynamic interactions amongst each other.

Despite the complex anatomy and facial soft tissue behavior, animators and artists require high fidelity face models that are easy to acquire, intuitive to control, and art-directable. To handle both complexity and art-directability, we propose hybrid deformation models, which are data-driven for simple parameter and model acquisition and utilize physically-based methods for editing and simulation.

In this thesis, we first investigate and model the geometry and motion of human faces. In particular, we focus both on large-scale motion, but also on motion on smaller scales such as expression wrinkles. We present techniques to efficiently acquire and model these features, introduce algorithms that allow real-time animation of highly detailed face models, and demonstrate that these subtle details contribute significantly to the expressiveness of virtual human faces. In addition, we investigate the soft tissue behavior and

introduce techniques for acquiring and simulating its deformation behavior. Based on this technique, we present a complete process for designing and even physical fabrication of deformable objects with complex heterogeneous materials using a multi-material 3D printer. Although fabricating artificial tissue with similar material properties than human skin is currently impractical due to limited capabilities of rapid prototyping machines, we think this work will pioneer future rapid prototyping for deformable objects and could be the basis for future facial animatronics and robotics.

1.1 Motivation

Realistic simulation behavior, surface details, and appearance are demanding tasks. Neither pure image-based, pure procedural, nor pure physical methods are best suited for accurate synthesis of facial motion and details (both for appearance and geometry), due to the difficulties in model design, parameter estimation, and desired controllability for animators. Capturing of a small but representative amount of real data, and then synthesizing diverse on-demand examples with procedural or physically-based models and real data as input benefits from both sides: Highly realistic model behavior due to real-world data and controllability due to physically-based models. The goal of this thesis is to easily acquire and represent 3D face models that can accurately simulate the behavior of a real person's face.

To model the face and its behavior, two different physically based approaches are elaborated, one using only surface representations such as thin shell models, and a second one using a volumetric representation based on finite elements. We introduce a three-dimensional face model that can accurately represent the different types of spatial and motion scales that are relevant for wrinkle modeling and animation. A central design element of our model is a decomposition of the facial features into fine, medium, and coarse spatial scales, each representing a different level of motion detail. This decomposition allows us to uniquely tailor the acquisition process to the spatial and temporal scale of expression wrinkle motions. The individual features at different scales are combined using a linear as well as a non-linear shell model. Furthermore, to simulate soft tissue in complex interaction scenarios, e.g. under external force loads, we introduce a volumetric data-driven representation and modeling technique for non-linear heterogeneous soft tissue that simplifies the construction of convincing deformable models.

To achieve realistic behavior, we propose to build light-weighted data capture devices to acquire real-world data to estimate model parameters and to

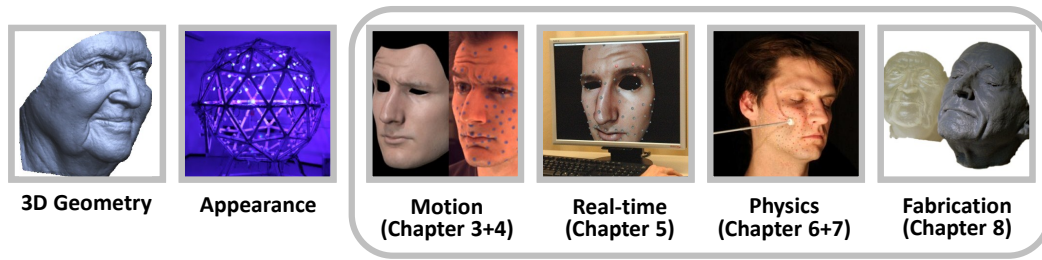


Figure 1.2: *Mastering and understanding human faces encompasses a variety of research challenges. This chart shows the most important ones. The scope of this thesis is on motion, focusing on expression wrinkles (Chapter 3), acquisition of face performances (Chapter 4), real-time animation and transfer of facial details (Chapter 5), underlying physics and deformation behavior (Chapter 7 + 6), and finally designing and fabricating materials with desired deformation behavior utilizing novel output devices (Chapter 8).*

employ concepts from data-driven modeling techniques. The resulting models support simple acquisition systems, provide techniques to process and extract model parameters from real-world data, feature a compact representation of the facial geometry and its motion, and allow intuitive editing. The proposed systems are repeatable and scalable.

Based on the volumetric data-driven model, we furthermore introduce a data-driven process for designing and fabricating materials with desired deformation behavior. The process starts with measuring deformation properties of base materials. Each material is represented as a non-linear stress-strain relationship in a finite-element model. For material design and fabrication, we introduce an optimization process that finds the best combination of stacked layers that meets a user’s criteria specified by example deformations. Our algorithm employs a number of strategies to prune poor solutions from the combinatorial search space. We demonstrate the complete process by designing and fabricating objects with complex heterogeneous materials using modern multi-material 3D printers.

1.2 Contributions

In this thesis we make the following contributions:

- ▶ A three-dimensional dynamic face model that can accurately repre-

sent the different types of spatial and motion scales that are relevant for wrinkle modeling and animation. A central design element of our model is a decomposition of the facial features into fine, medium, and coarse spatial scales, each representing a different level of motion detail. Medium-scale wrinkle geometry is added to the coarse-scale facial base mesh. Surface micro-structure, such as pores, is represented in the fine scale of the model.

- ▶ A tailored acquisition process to the spatial and temporal scale of expression wrinkle motions. We acquire a static high-resolution model of the face, including reflectance data. Then we place markers on the face and mark expression wrinkles with a diffuse color. We add two synchronized cameras to a marker-based optical motion-capture system and capture the facial performance. We adapt a linearized thin shell model to deform the high-resolution face mesh according to the captured motion markers. From the video data we estimate the expression wrinkles using a 2D parametric wrinkle model and add them to the deformed 3D face mesh by solving a non-linear energy minimization problem.
- ▶ A hybrid animation technique that combines computational and data-driven approaches, and thereby achieves detailed facial expressions that are intuitive to control as well as efficient to compute. Building on the previously proposed multi-scale decomposition, we compute the large-scale motion using the same linear shell deformation, but employ a data-driven approach for learning fine-scale facial details from a small set of example poses.
- ▶ A pose-space deformation (PSD) technique, which in a preprocess learns the correlation of wrinkle formation to sparsely measured skin strain. At run-time, given an arbitrary facial expression, it computes the skin strain and derives from it the necessary fine-scale corrections for the large-scale deformation.
- ▶ A novel data-driven representation and modeling technique for simulating non-linear heterogeneous soft tissue that simplifies the construction of convincing deformable models. Our technique employs finite element methods and exploits a set of measured example deformations of real-world objects, thereby avoiding complex selection of material parameters.
- ▶ An algorithm that transfers measured example deformation into a local element-wise strain space, and represent this example deformation as a locally linear sample of the material's stress-strain relation. We

1 Introduction

then model the full non-linear behavior by interpolating the material samples in strain space using radial basis functions (RBFs). Finally, a simple elastostatic finite-element simulation of the non-linearly interpolated material samples based on incremental loading allows for efficient computation of rich non-linear soft-tissue simulations.

- ▶ An algorithm and representation for coarsening deformable models with microscale inhomogeneous behavior. Our solution uses a data-driven approach and is capable of capturing non-linear behavior while maintaining runtime efficiency. Conceptually, this can be seen as an extension of the linear numerical coarsening approach by [Kharevych et al., 2009] to non-linear material behavior.
- ▶ A goal-based material design approach that approximates a desired mesoscale deformation behavior by microscale materials through combinatorial optimization. This is an inverse modeling approach, inverting the concept of homogenization.
- ▶ A *complete* reproduction process for deformable materials, including acquisition, fitting, efficient simulation, goal-based design, and fabrication. The result of our design process serves as input to a 3D multi-material printer for the actual physical fabrication of deformable objects.

1.3 Organization

The thesis is organized in three main parts. The first part investigates capturing and animating facial geometry and motion. In the second part, we introduce a volumetric model for soft tissue, and show in the third part how this model can be used for goal-based material design and fabrication.

In particular,

- ▶ **Chapter 2** presents a short introduction into fundamentals and related work.
- ▶ **Chapter 3** introduces a novel multi-scale representation for the animation of high-resolution facial geometry and wrinkles. The model consists of high-resolution geometry, motion-capture data, and expression wrinkles in 2D parametric form. During motion synthesis the motion-capture data deforms the high-resolution geometry using a linear shell-based mesh-deformation method. The wrinkle geometry is added to the facial base mesh using non-linear energy optimization.

- ▶ **Chapter 4** presents a simple acquisition method that is able to acquire all necessary parameters of the previously presented face model. We augment a traditional marker-based facial motion-capture system by two synchronized video cameras to track expression wrinkles and present an image-based algorithm for extracting coefficients of our wrinkle model from video data.
- ▶ **Chapter 5** builds on the previous chapters and presents a novel method for real-time animation of highly-detailed facial expressions based on a multi-scale decomposition of facial geometry into large-scale motion and fine-scale details. The hybrid animation is tailored to the specific characteristics of large- and fine-scale facial deformations: Large-scale deformations are computed with a fast linear shell model, which is controlled through a sparse set of motion-capture markers or user-defined handle points. Fine-scale facial details are incorporated using a novel pose-space deformation technique, which learns the correspondence of sparse measurements of skin strain to wrinkle formation from a small set of example poses. This hybrid method features real-time animation of highly-detailed faces with realistic wrinkle formation, and allows both large-scale deformations and fine-scale wrinkles to be edited intuitively. Furthermore, this pose-space representation enables the transfer of facial details to novel expressions or other facial models.
- ▶ **Chapter 6** presents data-capture devices for acquiring force-displacement pairs of deformable objects. They allow sampling and acquiring example deformations of the human face as well as other deformable objects.
- ▶ **Chapter 7** introduces a data-driven representation and modeling technique for simulating non-linear heterogeneous soft tissue. It simplifies the construction of convincing deformable models by avoiding complex selection and tuning of physical material parameters, yet retaining the richness of non-linear heterogeneous behavior. We acquire a set of example deformations of a real object, and represent each of them as a spatially varying stress-strain relationship in a finite-element model. We then model the material by non-linear interpolation of these stress-strain relationships in strain-space. Our method relies on a simple-to-build capture system and an efficient run-time simulation algorithm based on incremental loading, making it suitable for interactive computer graphics applications. We present the results of our approach for several non-linear materials and biological soft tissue, with accurate agreement of our model to the measured data.
- ▶ **Chapter 8** extends the data-driven representation presented in the pre-

1 Introduction

vious chapter and introduces a data-driven process for designing and fabricating materials with desired deformation behavior. The process starts with measuring deformation properties of base materials. Each material is represented as a non-linear stress-strain relationship in a finite-element model. For material design and fabrication, we introduce an optimization process that finds the best combination of stacked layers that meets a user's criteria specified by example deformations. Our algorithm employs a number of strategies to prune poor solutions from the combinatorial search space. We demonstrate the complete process by designing and fabricating objects with complex heterogeneous materials using modern multi-material 3D printers.

- ▶ **Chapter 9** summarizes the thesis and discusses the main contributions. Furthermore, it suggests potential future work.

1.4 Publications

This thesis is based on following accepted peer-reviewed publications:

- B. BICKEL, M. BÄCHER, M. A. OTADUY, H. R. LEE, H. PFISTER, M. GROSS, W. MATUSIK. Design and Fabrication of Materials with Desired Deformation Behavior. In *Proceedings of ACM SIGGRAPH (Los Angeles, USA, July 25-29, 2010)*, *ACM Transactions on Graphics*, vol. 29, no. 3.
- B. BICKEL, M. BÄCHER, M. A. OTADUY, W. MATUSIK, H. PFISTER, M. GROSS. Capture and Modeling of Non-Linear Heterogeneous Soft Tissue. In *Proceedings of ACM SIGGRAPH (New Orleans, USA, August 3-7, 2009)*, *ACM Transactions on Graphics*, vol. 28, no. 3.
- B. BICKEL, M. LANG, M. BOTSCH, M. A. OTADUY, M. GROSS. Pose-Space Animation and Transfer of Facial Details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Dublin, Ireland, July 7-9, 2008)*.
- B. BICKEL, M. BOTSCH, R. ANGST, W. MATUSIK, M. A. OTADUY, H. PFISTER, M. GROSS. Multi-Scale Capture of Facial Geometry and Motion. In *Proceedings of ACM SIGGRAPH (San Diego, USA, August 7-9, 2007)*, *ACM Transactions on Graphics*, vol. 26, no. 3.

During the time period of this thesis, but not directly related, following technical peer-reviewed papers were published:

- T. BEELER, B. BICKEL, R. SUMNER, P. BEARDSLEY, M. GROSS. High-Quality Single-Shot Capture of Facial Geometry. In *Proceedings of ACM SIGGRAPH (Los Angeles, USA, July 25-29, 2010)*, *ACM Transactions on Graphics*, vol. 29, no. 3.
- H.-J. KIM, B. BICKEL, M. GROSS, S.-M. CHOI. Subsurface scattering using splat-based diffusion in point-based rendering. In *SCIENCE CHINA Information Sciences, Science China Press and Springer-Verlag*, vol. 53, no. 5, 2010, pp. 911-919.
- N. PIETRONI, M. A. OTADUY, B. BICKEL, F. GANOVELLI, M. GROSS. Texturing Internal Surfaces from a Few Cross-Sections. In *Proceedings of Eurographics (Prague, Czech Republic, September 3-7, 2007)*, *Computer Graphics Forum*, vol. 26, no. 3, pp. 637-644.

1 Introduction

- B. BICKEL, M. WICKE, M. GROSS. Adaptive Simulation of Electrical Discharges. In *VMV 2006, Aachen, Germany, November 22-24, 2006*.
- T. WEYRICH, W. MATUSIK, H. PFISTER, B. BICKEL, C. DONNER, C. TU, J. MCANDLESS, J. LEE, A. NGAN, H. W. JENSEN, M. GROSS. Analysis of Human Faces using a Measurement-Based Skin Reflectance Model. In *Proceedings of ACM SIGGRAPH (Boston, USA, July 30 - August 2006)*, *ACM Transactions on Graphics*, vol. 25, no. 3.

Additional publications and book chapters during the time period of this thesis:

- B. BICKEL, M. LANG, M. BOTSCH, M. A. OTADUY. From sparse mocap to highly-detailed facial animation. Book chapter. To appear in *GPU Computing Gems, January 2011*.
- B. BICKEL, T. WEYRICH, W. MATUSIK, H. PFISTER, C. DONNER, C. TU, J. MCANDLESS, J. LEE, A. NGAN, H. W. JENSEN, M. GROSS. Implementation Sketch: Processing and Editing of Faces using a Measurement-Based Skin Reflectance Model. *Sketch ACM SIGGRAPH (Boston, USA, July 30 - August 2006)*.

Fundamentals and Related Work

This chapter presents general fundamentals and related work in the area of capturing and modeling facial geometry and soft tissue, as well as fabricating deformable materials. Face modeling, acquisition, and animation are rich areas of research in computer graphics [Noh and Neumann, 1999] and computer vision. In Section 2.1, we focus on related work in capturing static 3D models as well as facial performance capture. In Section 2.2, we give an overview of modeling techniques used for facial animation. Section 2.3 presents an overview of modeling complex elasticity properties. This goes beyond conventional animation techniques and allows physically-based simulations with external forces and complex boundary conditions. Finally, Section 2.4 introduces related work on material design and physical fabrication of objects based on novel output and fabrication devices such as computer-controlled milling machines and multi-material 3D printers.

2.1 Capture

Marker-Based Motion Capture The basic idea of combining 3D face geometry with marker-based motion-capture data dates back to [Williams, 1990]. Today, Vicon dominates the commercial market for marker-based facial-

2 Fundamentals and Related Work

capture systems, although many smaller companies and custom environments exist. These systems acquire data with excellent temporal resolution (up to 450 Hz), but due to their low spatial resolution (100-200 markers) they are not capable of capturing expression wrinkles.

Structured Light Systems Structured light techniques are capable of capturing models of dynamic faces in real time. [Zhang et al., 2004] use space-time stereo to capture face geometry, color, and motion. They fit a deformable face template to the acquired depth maps using optical flow. [Wang et al., 2004] use a sinusoidal phase-shifting acquisition method and fit a multi-resolution face mesh to the data using free-form deformations (FFD). [Zhang and Huang, 2006] improve this acquisition setup and achieve real-time (40 Hz) depth-map acquisition, reconstruction, and display. Structured light systems cannot match the spatial resolution of high-quality static face scans [Borshukov and Lewis, 2003, Sifakis et al., 2005] or the acquisition speed of marker-based systems. They also have difficulties in dealing with the concavities and self-shadowing that are typical for expression wrinkles.

Model-Based Animation from Video There has been a lot of work in fitting a deformable 3D face model to video (e.g., [Li et al., 1993, Essa et al., 1996, DeCarlo and Metaxas, 1996, Pighin et al., 1999]). Of special interest are linear [Blaiz et al., 2003] or multi-linear [Vlasic et al., 2005] morphable models that parameterize variations of human face geometry along different attributes (age, gender, expressions). Because these methods make use of some generic, higher level model, the reconstructed geometry and motion do not approach the quality of person-specific captured data. [Hyneman et al., 2005] compensated the lack of details by adding a dynamic displacement map that included hand-painted wrinkles and furrows.

Image-Based Methods with 3D Geometry [Guenter et al., 1998] and [Borshukov et al., 2003] compute a time-varying texture map from multiple videos and apply it to a deformable face model fitted to the video. [Jones et al., 2006] use the USC Light Stage [Wenger et al., 2005] augmented with a high-speed camera and projector to capture the reflectance field and 3D geometry of a face. They relight the face using the time-varying reflectance data and simulate spatially-varying indirect illumination. Image-based methods are able to produce the most photo-realistic examples of facial performance. However, they typically lack in versatility with respect to editing and changes in head pose and illumination. In principle it should be possible to combine our approach with an image-based method.

Anatomical Face Models Anatomical models provide an animator with model parameters that have bio-mechanical meaning [Koch et al., 1996, Magnenat-Thalmann et al., 2002]. Some models were specifically developed

2.1 Capture

for wrinkles [Wu et al., 1996, Zhang and Sim, 2005, Venkataraman et al., 2005], but have not been applied to facial motion capture. To simulate wrinkle bulges due to facial expressions, we have found that it suffices to model the skin as a non-linear shell resistant to stretching and bending [Grinspun et al., 2003, Bridson et al., 2003]. [Terzopoulos and Waters, 1993] tracked marks on a performer’s face using snakes, and used these curves to drive a muscle-based facial model. [Sifakis et al., 2005] developed a highly detailed anatomical face model and morph it to fit laser and MRI scans of a new subject. They use sparse marker-based motion-capture data to automatically determine muscle activations. The face mesh is deformed using a 3D non-linear finite element method. However, generic anatomical face models are currently not able to produce expression wrinkles for an individual.

Normal-Estimation Methods with 3D Geometry [Nehab et al., 2005] showed that normal and positional information can be efficiently combined to obtain high-resolution geometry. [Weyrich et al., 2006] use photometric stereo, and [Ma et al., 2007] polarized gradient-based illumination to obtain high-resolution normals to augment 3D geometry. Recently, in combination with marker-based motion capture, this approach was extended to dynamic performance capture [Ma et al., 2008, Klaudiny et al., 2010]. [Ma et al., 2008] builds on our hybrid approach of combining physically-based and data-driven techniques but captures even pore-level deformations with time-multiplexed gradient illumination.

Passive Systems without Markers do not require any active illumination, special markers, or face paint. Passive stereo systems usually cannot resolve spatial features at pore level given the resolution of current HD cameras. [Beeler et al., 2010] overcomes this limitation by introducing an image-based embossing technique to capture mesoscopic facial geometry. [Bradley et al., 2010] achieves dense passive performance capture by segmenting the face into several patches. For each patch, a pair of zoomed in HD video cameras provides sufficient resolution to perform stereo matching and a surface tracking based on optical flow. While this approach is data-intensive and computationally expensive, in principle it should be possible to combine it with our hybrid modeling approach to provide a more compact model and extended editing capabilities.

2.2 Modeling for Animation

Modeling and animation of human faces are active research areas since the early days of computer graphics. Rapidly increasing computation power, high-resolution, precise sensors, and sophisticated performance capture algorithms provide the basis for further understanding human faces and have led to a variety of modeling and animation techniques. Our proposed method is kindred to techniques in character skinning, hence we focus our discussion of related work on those research areas.

Blending One large family of methods in face animation employs model blending. Blend shapes, dating back to the early work of Parke [Parke, 1974] and commonly used in the industry today, linearly combine several poses of the same face model, either designed by an artist or acquired from a person. They are often used in combination with the Facial Action Coding System (FACS) developed by [Ekman and Friesen, 1978] to create facial rigs. The original FACS provides a study of different basic expressions, what muscles control them, and a set of in total 46 action units and action descriptors that allow to encode a wide range of facial expressions. In practice, often many additional poses are required to sufficiently cover the space of facial expressions. In general, linear blending suffers from two main drawbacks: It requires a large number of poses for spanning the range of possible expressions [Fordham, 2003], and blending controls are hard to tune. Blend shapes have therefore been extended to automatic blending control [Pighin et al., 1999] and automatic segmentation [Joshi et al., 2003], but those methods require computationally expensive optimizations. The FaceIK technique of [Zhang et al., 2004] allows for interactive and local blending control when the number of matched points is smaller than the number of blend shapes. Morphable models [Banz and Vetter, 1999] and their extension to multilinear models [Vlasic et al., 2005] not only blend different expressions from the same object, but also from a large database of different subjects. The common drawback of blend shapes, however, is that they typically do not allow for intuitive, precise, and interactive control at the same time.

Anatomical Models Another family of methods uses anatomical models activated by controls with biomechanical meaning [Koch et al., 1996, Essa et al., 1996, Sifakis et al., 2005]. They allow extending the range of motions beyond muscle-driven expressions by incorporating external forces. [Essa et al., 1996] learn muscle activations for matching video-captured poses, and then interpolate the learned poses. [Sifakis et al., 2005] learn functions for muscle activation in a more sophisticated tissue model. Furthermore, several approaches focus on specific anatomical models for

wrinkle simulation [Magnenat-Thalmann et al., 2002, Zhang and Sim, 2005, Venkataraman et al., 2005]. Among those, [Wu et al., 1996] designed procedural methods for wrinkle synthesis as a function of the strain measured in an underlying tissue model. Those approaches allow for realistic model behavior, but in turn require complex parameter tuning and expensive computations.

The parameter-setting complexity of anatomical models can be largely relaxed by adding user- or data-driven constraints to the deformation. [Pighin et al., 1998] matched corresponding vertices in a generic model to mocap markers, and then smoothly interpolated the deformation on the rest of the face. In Chapter 3 and [Bickel et al., 2007], we successfully decompose face animations into two scales, computing the large-scale motions by a fast linear shell model driven by mocap markers, thus requiring an expensive non-linear model only for synthesizing fine-scale wrinkles. Furthermore, we require spatial and temporal dense capturing of geometric wrinkle constraints. We therefore extend this model in Chapter 5 and propose a hybrid framework that exploits the quasi-static nature of fine-scale wrinkles by employing an example-based model for fine-scale skin corrections, and thereby avoids spatio-temporal dense acquisition and any expensive non-linear optimizations at run-time.

Example-based skin correction models have been successfully used for modeling arms [Lewis et al., 2000] or hands [Kry et al., 2002, Kurihara and Miyata, 2004]. They typically combine a fast, linear skeletal subspace deformation (SSD) [Magnenat-Thalmann et al., 1988] with a non-linear pose-space deformation (PSD) [Lewis et al., 2000] that interpolates correction vectors among example poses. PSD was extended to support weighted (i.e., per-vertex) pose space deformation (WPSD) [Kurihara and Miyata, 2004, Rhee et al., 2006], which largely reduces the number of required example poses. Similar in spirit to PSD, the EigenSkin method [Kry et al., 2002] performs corrections to SSD, but derives a reduced basis from a PCA of the input examples. The recently presented method of [Meyer and Anderson, 2007] is related to our work in the sense that it computes high-resolution deformations from a few handle vertices, but it focuses on the selection of handles given a large training dataset. Other recent methods [Wang et al., 2007, Weber et al., 2007] learn example-based corrections on sparse points and assume that these corrections can be smoothly interpolated. In general, any pose-space method requires the definition of a suitable pose space, which for SSD can easily be defined using bone transformations. In the context of face models, we define a novel pose space based on local skin strain and derive corresponding PSD and WPSD techniques for fine-scale skin correction.

Transfer Several previous approaches address the transfer of whole face animations from one subject onto other faces [DeCarlo and Metaxas, 2000, Noh and Neumann, 2001, Blanz et al., 2003, Chai et al., 2003] as well as [Sumner and Popović, 2004] and more recently [Weise et al., 2009, Li et al., 2010a]. In contrast, as we will demonstrate in Chapter 5, our pose-space representation allows to transfer only the fine-scale details from one character onto another, which can be used to add realistic wrinkles to existing models and face animations. Notably, [Golovinskiy et al., 2006] presented a statistical model for synthesizing detailed facial geometry, but it is limited to a single static expression and cannot handle visually important expression wrinkles.

2.3 Modeling Soft Tissue

Researchers in many fields, ranging from mechanical engineering to biology, have long studied the problem of modeling complex elasticity properties. For a recent survey of deformation models in computer graphics, please refer to [Nealen et al., 2006].

Bio-Mechanical Models For soft tissue modeling, a common approach is to devise a constitutive model [Ogden, 1997] that captures in a sufficiently accurate manner the various behavior regimes of the material, and then tune the model parameters until they best fit empirical data. This approach is, however, tedious and utterly complex, as it relies on accurate modeling of tissue geometry (e.g., the bones, fat, and muscles in facial tissue), rich excitation of material regimes, and accurate measurement of forces and deformations (even in typically inaccessible regions). Despite the complexity of the approach, it has seen large application in computer graphics since the pioneering work by [Terzopoulos et al., 1987], as it can lead to stunning results with the appropriate amount of effort. Some examples of complex bio-mechanical models in computer graphics include the neck [Lee and Terzopoulos, 2006], the torso [Zordan et al., 2004, Teran et al., 2005, DiLorenzo et al., 2008], the face [Koch et al., 1996, Magnenat-Thalmann et al., 2002, Terzopoulos and Waters, 1993], even controlled by muscle activations from motion capture data [Sifakis et al., 2005], and the hand [Sueda et al., 2008].

Measurement-Based Model Fitting To circumvent the complexity of parameter tuning, several authors have proposed measurement-based model fitting approaches. The seminal work of [Pai et al., 2001] presents a capture and modeling system for a deformable object's shape, elasticity, and

surface roughness. Their deformable model was based on a Green's functions matrix representation [James and Pai, 1999], and was later extended to increase fitting robustness [Lang et al., 2002], and to handle viscoelasticity [Schoner et al., 2004]. Our approach introduced in Chapter 7 shares their strategy for measuring surface displacements as the result of applied surface forces, but, unlike theirs, is not limited to linear material behavior and does not rely on global response functions. [Sifakis et al., 2005] give a different spin to measurement-based modeling approaches, as they learn the relationship between facial muscle activation and skin positions. Others, particularly in biomechanics, have explored measurement-based fitting of the parameters of various constitutive models, such as Young modulus estimation based on a non-linear least squares problem [Schnur and Zabararas, 1992], Young modulus and Poisson ratio estimation through linear least squares [Becker and Teschner, 2007], estimation of non-linear viscoelastic materials [Kauer et al., 2002], or even plasticity estimation [Kajberg and Lindkvist, 2004]. Our work borrows from these approaches for the estimation of each individual sample of the stress-strain relationship. However, this alone is not sufficient for capturing the rich non-linear behavior of soft tissue. In contrast to previous work, the realism of our material model is greatly enhanced with spatially varying non-linear interpolation in strain space.

Data-driven Methods Purely data-driven techniques have gained large popularity in computer graphics, as they may produce highly realistic results for phenomena that are otherwise extremely complex to model. The interpolation of lightfield samples [Buehler et al., 2001] allows simulating the illumination of complex scenes, while recent data-driven reflection models [Matusik et al., 2003] represent each BRDF through a dense set of measurements. Data-driven methods have also been applied to several other aspects of deformation modeling in computer graphics, such as facial wrinkle formation from local skin deformations [Ma et al., 2008, Bickel et al., 2008], grasping of objects [Kry and Pai, 2006], skeleton-driven cloth wrinkles [Kim and Vendrovsky, 2008], body-skin deformation [Park and Hodgins, 2006], or learning of skeleton-driven skin dynamics [Park and Hodgins, 2008]. Our method is a mixture of model fitting techniques (i.e., estimating stress-strain parameters from local measurements) and data-driven methods (i.e., using tabulated stress-strain parameters and non-linear interpolation during runtime).

Shape Modeling Another common approach in computer graphics to model deformations is shape modeling [Botsch and Sorkine, 2007]. Some of the existing approaches rely on predefined examples [Sloan et al., 2001, Allen et al., 2002, Sumner et al., 2005], or even exploit interpolation

2 Fundamentals and Related Work

[Bergeron and Lachapelle, 1985, Lewis et al., 2000, Blanz et al., 2003], but these techniques cannot model deformations as a reaction to contact in the way our technique does. Some recent approaches connect shape modeling with physically-based reactive models, by rigging using templates of forces [Capell et al., 2005] or by skeletal interpolation of elastic forces [Galoppo et al., 2009], but these approaches cannot model general non-linear soft tissue.

Deformation Capture Our work captures deformation examples by combining a stereo-vision acquisition system and force sensors, similar to earlier approaches [Pai et al., 2001]. Other techniques in material sciences also directly measure the parameters of constitutive models, such as the tensile test [Hart, 1967], or apparatus for in-vivo measurement through tissue aspiration [Nava et al., 2003] or indentation [Ottensmeyer and Salisbury Jr., 2004].

2.4 Material Design and Fabrication

Elastic deformations are present in many objects in our everyday life, such as our face and own tissue, but also in garments and shoes, furniture, or plants.

Recent development of multi-material 3D printers such as the OBJET Connex series [OBJET, 2009] are capable of manufacturing a variety of soft and hard materials with complex internal structures, making it suddenly possible to fabricate complex 3D objects with aggregate materials quickly, inexpensively, and accurately. This development offers new opportunities and challenges for physics-based animation research.

Multi-scale Physics Deformation effects can be modeled at very diverse scales, ranging from molecular interactions to globally-supported response functions, and through continuum elasticity laws or lumped-parameter models [Zohdi and Wriggers, 2004].

High-resolution Heterogeneities Recent work in computer graphics aims at modeling high-resolution heterogeneities even when the resolution of the discretization is considerably coarser [Kharevych et al., 2009, Nesme et al., 2009]. This process, known as *homogenization*, tries to find parameter values of a constitutive model sampled at low resolution such that the behavior of the object best matches the heterogeneous material. The first step of our design and fabrication process introduced in Chapter 8 can be considered as a variant of homogenization, where the fine-scale inhomogeneous material is an actual physical one. In the second step, however, we

take the opposite approach to homogenization, generating a heterogeneous object that fits coarse force-deformation data from small-scale materials with known behavior.

A general introduction to optimizing spatial material distributions can be found in [Bendsoe and Sigmund, 2003]. Digital materials, composed of a set of discrete voxels, can exhibit widely varying material properties [Hiller and Lipson, 2009]. As the design space increases exponentially with the number of possible combinations, evolutionary algorithms are a popular non-linear optimization strategy [Kicinger et al., 2005]. In contrast, we apply a branch-and-bound search strategy in combination with clustering.

Replicating Reality Given the general challenge of replicating reality, a wide range aspects has to be addressed. Among the most important ones are *3D geometry*, where several techniques exist to both capture (e.g. [Levoy et al., 2000]) and also fabricate static objects [Liou, 2007]. Another aspect is *appearance*, where [Matusik et al., 2009] recently presented an approach for printing spatially-varying reflectance, and [Weyrich et al., 2009] showed an approach for fabricating microgeometry for custom surface reflectance. Furthermore, recently approaches have been presented for physical fabrication of objects with desired subsurface scattering [Hasan et al., 2010, Dong et al., 2010]. However, there is a lack of tools and algorithms for designing, editing, and fabricating objects with desired *deformation behavior*. Based on the material acquisition system described in Chapter 6 and the modeling techniques for soft tissue presented in Chapter 7, we present a novel complete system for design and fabrication of deformable objects.

Notably, in the wider context, computer graphics has already contributed systems for designing and fabricating clothes [Okabe et al., 1992], plush objects [Mori and Igarashi, 2007, Igarashi and Igarashi, 2009], and paper craft objects [Mitani and Suzuki, 2004, Li et al., 2010b].

Modeling Facial Geometry and Motion

As humans are especially good at detecting and recognizing subtle facial expressions, a central part of this thesis is to present a compact representation that is able to sufficiently cover both the *dynamics* of the expression as well as *detailed spatial deformations*.

Although the movie industry continues to make steady progress in digital face modeling, current facial capture, modeling, and animation techniques are not able to generate an adequate level of spatio-temporal detail without substantial manual intervention by skilled artists. Our goal is to represent 3D face models that can accurately animate the spatial and temporal behavior of a real person's facial wrinkles. While developing this representation, an essential design requirement was that all parameters can be easily and automatically acquired and estimated from a real person's face.

In this chapter, we introduce a multi-scale representation for of high-resolution facial geometry and wrinkles, followed by Chapter 4 that presents an efficient and simple performance capture method that is able to acquire all necessary parameters. Section 4.4 shows results of our approach for performance replay as well as for wrinkle editing, and in Chapter 5 we demon-

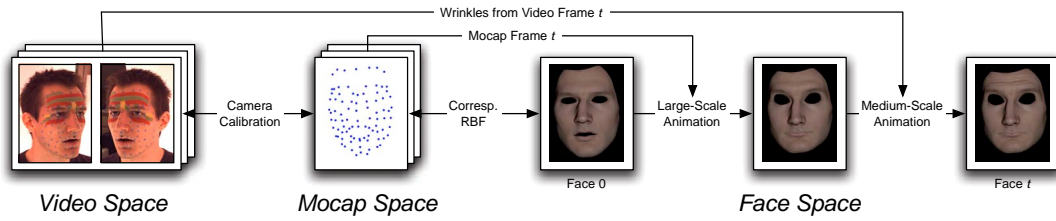


Figure 3.1: Our multi-scale face model first computes a large-scale linear deformation, on top of which medium-scale wrinkles are synthesized. In our framework we capture a video sequence and motion capture markers of an actor’s performance, together with a static, high-resolution face scan (Chapter 4). The camera calibration and correspondence function enable the transfer of information between those spaces.

strate that this representation is furthermore suited for real-time animation and wrinkle transfer.

3.1 Motivation

Facial skin can be represented by a hierarchy of skin components based on their geometric scale and optical properties [Igarashi et al., 2005]. In the visible domain, they range from the fine scale (e.g., pores, moles, freckles, spots) to the coarse scale (e.g., nose, cheeks, lips, eyelids). Somewhere between those scales are expression wrinkles that occur as a result of facial muscle contraction [Wu et al., 1996]. We call this hierarchy the *spatial scales* of the face.

Facial motion can also be characterized at multiple time scales. At the short-time, high-frequency end of the scale are subtle localized motions that can occur in a fraction of a second. More global motions, such as the movement of the cheeks when we speak, are somewhat slower. And at the smallest spatial scale, features such as pores or moles hardly show any local deformations and can be considered static in time. Expression wrinkles are somewhere between those extremes. They can occur quickly, but they do not move fast during facial expressions (e.g., try moving the wrinkles on your forehead quickly). We call this hierarchy the *motion scales* of the face.

In this chapter we present a three-dimensional dynamic face model that can accurately represent the different types of spatial and motion scales that are relevant for wrinkle modeling and animation. A central design element of

3.2 Large-Scale Animation

our model is a *decomposition of the facial features into fine, medium, and coarse spatial scales, each representing a different level of motion detail*. Medium-scale wrinkle geometry is added to the coarse-scale facial base mesh. Surface microstructure, such as pores, is represented in the fine scale of the model. This decomposition allows us to uniquely tailor the acquisition process to the spatial and temporal scale of expression wrinkle motions.

The conceptual components of our facial-capture approach and representation are illustrated in Figure 3.1. First we acquire a static high-resolution model of the face, including reflectance data. Then we place approximately 80–90 markers on the face and mark expression wrinkles with a diffuse color. We add two synchronized cameras to a marker-based optical motion-capture system and capture the facial performance. We adapt a linearized thin shell model to deform the high-resolution face mesh according to the captured motion markers. From the video data we estimate the expression wrinkles using a 2D parametric wrinkle model and add them to the deformed 3D face mesh by solving a non-linear energy minimization problem.

Decomposing the face model into these separate components has several advantages. Throughout the animation, the face geometry maintains the high-resolution of the static scan and preserves a consistent parameterization for the texture and reflectance data. In addition, the face mesh maintains dense correspondence throughout the animation, so that edits on the geometry, textures, and reflectance parameters are automatically propagated to each frame. The model is compact and provides data in a form that is easy to edit.

The primary contribution of our work presented in this chapter is the multi-scale facial representation for the animation of expression wrinkles. This model, which is practical and easy to use, allows for the decomposition of the capture process for dynamic faces into fine, medium, and coarse components. The model includes a variety of computational steps for the mapping of motion-capture data, facial deformation, and wrinkle animation.

The versatility of our model including results for performance replay and wrinkle processing are presented in Section 4.4.

3.2 Large-Scale Animation

A simple representation of large-scale motion is a sparse set of marker positions in the face and its displacements over time. In fact, marker-based optical motion capture is the most common commercially used technology to acquire coarse facial motions at frame rates up to 2000 Hz (e.g., www.vicon.com).



Figure 3.2: Animation of a high-resolution face scan using marker-based motion capture and a video-driven wrinkle model. From left to right: video frame, large-scale animation without wrinkles, synthesis of medium-scale wrinkles, realistic skin-rendering, different expression.

The motion-tracking process results in a set of time-dependent marker positions $\mathbf{m}_{i,t} \in \mathbb{R}^3$, $i = \{1, \dots, n\}$, $t = \{0, 1, \dots\}$ in the reference space of the motion-capture system (*mocap space*). At a certain time t , the difference vectors $(\mathbf{m}_{i,t} - \mathbf{m}_{i,0})$ represent point-samples of the continuous deformation field that deforms the initial face model into the expression at frame t . Our goal is to deform an initial face mesh F based solely on these displacement constraints.

Since the 3D scan F and the mocap points are defined with respect to different coordinate systems, the points $\mathbf{m}_{i,0}$ and their respective displacements $(\mathbf{m}_{i,t} - \mathbf{m}_{i,0})$ first have to be mapped to the coordinate space of the face mesh F (*face space*), resulting in points $\mathbf{f}_{i,0}$ and displacements $\mathbf{u}_{i,t} = (\mathbf{f}_{i,t} - \mathbf{f}_{i,0})$. We achieve this by establishing a correspondence function as described in Section 4.2.1.

The resulting displacements $\mathbf{u}_{i,t}$ in face space are then used as constraints for our physically inspired face deformation model. Notice that a physically accurate face deformation — including the interaction of bones, muscles, and tissue — is too complex for our purposes. From our experiments it turned out that the mocap points capture the *large-scale* face behavior sufficiently well, so that we can use a simplified deformation model that interpolates the mocap points (see Section 3.2.1).

3.2.1 Linear Deformation Model

After transferring the mocap displacements into face space, we deform the initial face mesh based on these displacement constraints. This requires a

3.2 Large-Scale Animation

deformation function $\mathbf{u}_t : F \rightarrow \mathbb{R}^3$ that is smooth and physically plausible while interpolating the constraints of frame t :

$$\mathbf{u}_t(\mathbf{f}_{i,0}) = \mathbf{u}_{i,t}, \quad \forall i = 1, \dots, n, \quad (3.1)$$

such that $\mathbf{f}_{i,0} + \mathbf{u}_t(\mathbf{f}_{i,0}) = \mathbf{f}_{i,t}$. Note that another RBF-like space deformation is not suitable, since the desired deformation might be discontinuous around the mouth and eyes, whereas an RBF would always yield a C^2 continuous deformation.

For the global *large-scale* face deformation we propose using a *linear* shell model, since this allows for efficient as well as robust animations, even for our complex meshes of about 700k vertices. The missing medium-scale non-linear effects, i.e., wrinkles and bulges, are added later as described in Section 3.3.1.

Our linearized shell model incorporates the prescribed displacements $\mathbf{u}_{i,t}$ as boundary constraints, and otherwise minimizes surface stretching and bending. After linearization, the required stretching and bending energies can be modeled as integrals over first- and second-order partial derivatives of the displacement function \mathbf{u}_t [Celniker and Gossard, 1991]:

$$\int_F k_s \underbrace{\left(\left\| \frac{\partial \mathbf{u}_t}{\partial u} \right\|^2 + \left\| \frac{\partial \mathbf{u}_t}{\partial v} \right\|^2 \right)}_{\text{stretching}} + k_b \underbrace{\left(\left\| \frac{\partial^2 \mathbf{u}_t}{\partial^2 u} \right\|^2 + 2 \left\| \frac{\partial^2 \mathbf{u}_t}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial^2 \mathbf{u}_t}{\partial^2 v} \right\|^2 \right)}_{\text{bending}} du dv. \quad (3.2)$$

The deformation \mathbf{u}_t that minimizes this energy functional can be found by solving its corresponding Euler-Lagrange equations

$$-k_s \Delta \mathbf{u}_t + k_b \Delta^2 \mathbf{u}_t = 0 \quad (3.3)$$

under the constraints of Equation (3.1). Since our displacement function \mathbf{u}_t is defined on the initial mesh F , i.e., on a triangulated two-manifold, Δ represents the discrete Laplace-Beltrami operator. It can be discretized using following form as defined in [Meyer et al., 2003]:

$$\Delta \mathbf{u}_{t,i} = w_i \sum_{v_j \in N_1(v_i)} w_{ij} (\mathbf{u}_{t,j} - \mathbf{u}_{t,i}), \quad (3.4)$$

where $\mathbf{u}_{t,i}$ is the displacement of the i 'th vertex v_i at frame t , and $v_j \in N_1(v_i)$ are its incident one-ring neighbors. For the per-vertex normalization weights and the edge weights we are using the de-facto standard cotangent discretization

$$w_i = 1/A_i, w_{ij} = 1/2(\cot \alpha_{ij} + \cot \beta_{ij}), \quad (3.5)$$

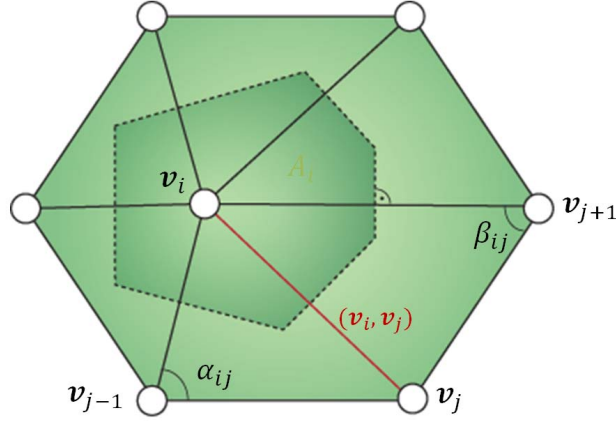


Figure 3.3: *The cotangent discretization [Meyer et al., 2003]. The weights of the Laplacian operator Δ at a vertex v_i are computed by considering the Voronoi area A_i and the edge weights w_{ij} based on the cotangents of the angles α_{ij} and β_{ij} opposite to the edge.*

where α_{ij} and β_{ij} are the two angles opposite to the edge (v_i, v_j) , and A_i is the Voronoi area of vertex v_i as shown in Figure 3.3.

With this discretization, the above PDE leads to a sparse linear system to be solved for the displacements at all mesh vertices, similar to [Botsch and Kobbelt, 2004]. Notice, however, that in contrast to the latter paper, we compute a smooth deformation field instead of a smooth surface. As a consequence, all small-scale details of F , such as pores and fine aging wrinkles, are retained by the deformation.

This linear system has to be solved for every frame of the mocap sequence, since each set of transferred mocap displacements $\mathbf{u}_{i,t}$ yields new boundary constraints, i.e., a new right-hand side. Although the linear system can become rather complex — its dimension is the number of free vertices — it can be solved efficiently using either a sparse Cholesky factorization or iterative multigrid solvers [Botsch et al., 2005, Shi et al., 2006]. All animations in this and the following chapter were computed with the parameters $k_s = 1$ and $k_b = 100$.

Since the global face motion does not contain significant local rotations, there is no need to explicitly rotate small-scale details, e.g., by multi-resolution decomposition or differential coordinates [Botsch and Sorkine, 2007]. Although the deformation of the human face is the result of complex interactions between skull, muscles, and skin tissue, the linear deformation model yields visually plausible results because the motion-capture markers provide

sufficient geometric constraints. While the resulting animations are of high visual quality, non-linear effects such as expression wrinkle formation obviously cannot be produced by the linearized deformation model. The next section describes how we enhance the large-scale facial animation with medium-scale expression features that can for example be extracted from video data as demonstrated in Section 4.3.

3.3 Medium-Scale Animation

In this section, we describe a physically-inspired *non-linear* shell deformation model that allows us to synthesize *medium-scale* 3D expression wrinkles and bulging onto the large-scale animation. To control the wrinkle formation, it requires as input the location and opening angle of the wrinkle valleys, which can be easily obtained by an image-based algorithm as described in Section 4.3.

Skin is a multilayer, anisotropic, viscoelastic tissue, whose mechanical behavior is dominated by collagen fibers present in the dermis [Lanir, 1987]. Hence, accurate simulation of skin folding would require a complex volumetric representation with carefully chosen model parameters [Magenat-Thalmann et al., 2002, Sifakis et al., 2005]. For the purpose of simulating wrinkle bulge formation due to facial expressions, however, we found our non-linear shell model to be sufficient.

3.3.1 3D Wrinkle Synthesis

In Section 3.2.1 we employed a *linear* shell model for the *large-scale* face animation. We now refine this result by synthesizing *medium-scale* wrinkles onto the large-scale facial animation based on a *non-linear* shell energy minimization.

We employ the non-linear discrete shell energy of [Grinspun et al., 2003] to measure the difference between the initial mesh F and its deformed version. Their energy is defined in terms of geometric quantities of the triangle mesh,



Figure 3.4: The synthesis of medium-scale wrinkles start from the large-scale linearly deformed mesh (left), on top which wrinkle valleys are added as normal displacements, based on the projected wrinkle functions extracted from the video (center). Our non-linear minimization of surface stretching and bending finally gives the missing bulging between neighboring wrinkles (right).

and measures the change of edge lengths $\|e_i\|$, complements of the dihedral angles θ_i , and triangle areas $\|t_i\|$, over all edges e_i and triangles t_i .

$$E_{\text{shell}} = \sum_{e_i} k_e \cdot \frac{(\|e_i\| - \|\bar{e}_i\|)^2}{\|\bar{e}_i\|} + \sum_{e_i} k_b \cdot \frac{\|\bar{e}_i\| (\theta_i - \bar{\theta}_i)^2}{\bar{h}_i} \quad (3.6)$$

$$+ \sum_{t_i} k_a \cdot \frac{(\|t_i\| - \|\bar{t}_i\|)^2}{\|\bar{t}_i\|},$$

where the “barred” terms $\|\bar{e}_i\|$, $\|\bar{t}_i\|$, and $\bar{\theta}_i$ denote the edge length, triangle area, and dihedral angle in the undeformed rest state F . The angle weighting by edge length $\|\bar{e}_i\|$ and triangle height \bar{h}_i accounts for irregular triangulations [Grinspun et al., 2003].

The geometric constraints for the non-linear energy minimization are constituted by the locations and cross-section profiles of the wrinkle valleys extracted from video data. The cross-section profiles are represented as a spatially varying wrinkle shape function $S(p)$, where p is the distance orthogonal to the wrinkle. We map the linearly deformed face F into mocap space using the inverse correspondence map \mathbf{c}^{-1} , project all 2D wrinkle splines $\mathbf{v}(x)$ onto it based on the camera parameters, and map the result back to face space using \mathbf{c} . The detailed description on how we establish the correspondence map \mathbf{c} and obtain the shape function $S(p)$ from video can be found in Section 4.2.1 and Section 4.3, respectively.

Then, we evaluate the wrinkle shape function $S(p)$ in the valley $-w < p < w$ for all cross-sections x along the spline $\mathbf{v}(x)$, and offset the affected mesh vertices along their (smoothed) normals (see Section 3.3.2). This procedure

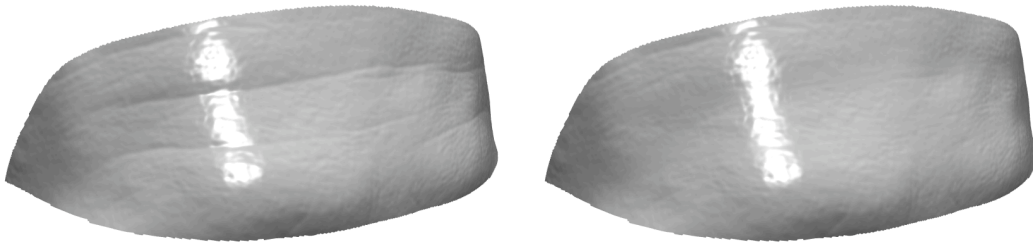


Figure 3.5: *In a preprocess we remove wrinkles that already exist in the initial, relaxed-pose face scan (left). Our multi-scale smoothing eliminates medium-scale wrinkles, but at the same time preserves large-scale geometry as well as small-scale details (right).*

provides absolute positions for vertices corresponding to the wrinkle valley. As explained in more detail in Section 4.3, we work with two cameras in order to cover the whole facial area. If a wrinkle is tracked by both cameras, we merge the detected segments by simple snapping and linear blending.

With wrinkle valleys constituting the geometric constraints, we perform a minimization of the shell energy Equation (3.6). This updates the vertices of the face mesh, such that surface area and curvature of the initial scan F are approximately preserved, which then leads to the required bulging between neighboring wrinkles.

We solve the minimization problem, and thus compute the final face animation, using Gauss-Newton optimization. We initialize vertex positions at the configuration obtained by the large-scale linear deformation. All animations in this and the following chapter were rendered with the parameters $k_b = 2$, $k_e = 300$, and $k_a = 30k$. As we are dealing with high-resolution meshes, a global Gauss-Newton optimization would be computationally too expensive. Therefore, as a heuristic, we determine the influence region of the wrinkles by using a predefined maximum influence distance. By merging overlapping regions, we obtain an automatic segmentation of the face into wrinkling and non-wrinkling areas. For each wrinkling area, we optimize (3.6) independently, while keeping the remaining vertices fixed.

3.3.2 Wrinkle Removal

The framework presented so far detects wrinkles from video data, projects them onto the linearly deformed face mesh, and recovers bulges between wrinkles by a non-linear energy minimization. If the initial face scan (relaxed

3 Modeling Facial Geometry and Motion

pose) already contains noticeable wrinkles, however, they would be detected and erroneously amplified by our technique. We therefore remove existing wrinkles from the static face scan in a preprocess.

We employ a three-step multi-scale smoothing to wrinkle regions in order to remove only the medium-scale wrinkles, but preserve both the large-scale face geometry and the small-scale details.

1. First we subtract the fine-scale details by a small amount of Laplacian smoothing [Desbrun et al., 1999], and store them as local-frame displacements [Kobbelt et al., 1999].
2. We eliminate the medium-scale wrinkles by minimizing curvature energies. This is equivalent to computing the steady state of bi-Laplacian smoothing [Desbrun et al., 1999], but only requires solving a bi-Laplacian linear system.
3. The resulting surface patch is smooth and blends with the surrounding non-wrinkle mesh in a tangent-continuous manner. Consequently, it preserves the global, large-scale geometry. On top of this smooth patch we finally add back the fine-scale details as normal displacements to get the desired result.

The effect of this multi-scale smoothing is depicted in Figure 3.5.

3.4 Discussion

In this chapter, we presented a compact representation for facial geometry and motion. Fine-scale details such as pores are represented in a single high-resolution 3D surface mesh, medium-scale details such as expression wrinkles are encoded as a simple temporally varying parametric model based on measurements, and large-scale motion is encoded as displacements of a small set of marker positions. The actual facial expression is computed using physically-based deformation methods.

In the next chapter we will show that with a simple performance capture system and an image-based algorithm all necessary parameters of the model can be obtained easily.

By design, our model is suited only for performance capture and replay. We therefore extend the model in Chapter 5, introducing an interpolation technique that allows to synthesize facial details for expressions that were not directly captured.

In its current form it does not provide intuitive parameters for animation control. While animators could directly control for example marker positions or the parameters of wrinkles, more high-level interfaces with semantic meaning, controls of muscle activations [Kähler et al., 2001], visemes [Ezzat and Poggio, 2000], or emotional states [Ostermann, 1998] are outside the scope of the thesis. However, in theory one could combine those approaches with our representation.

A further limitation of our model is its lack of facial anatomy and physics. This includes eyes and teeth, but also skin and muscle layers, or self-collision. If needed, such features could be imported by combining our model with other existing ones, such as [Sifakis et al., 2005].

Finally, we are interested in wrinkle animation transfer between individuals, an issue of high practical relevance for applications in the special-effects industry. Chapter 5 will address this issue, and will show how this model can be applied and extended for efficient real-time animation and transfer of facial details.

Performance Capture

In this chapter we present a performance capture system for three-dimensional dynamic face models that can accurately acquire the different types of spatial and motion scales that are relevant for wrinkle modeling and animation. Its design is tailored to the representation of facial geometry and motion presented in the previous chapter, and the acquired data is an essential part for further experiments on animation and transfer of facial details, as presented in Chapter 5.

4.1 Overview

Our face model presented in the previous chapter consists of three components, representing low, medium, and high-frequency features at different temporal resolutions. While there are several existing approaches to either scan a high-resolution, static face model or acquire motion at a relatively sparse spatial sampling using motion capture markers, the acquisition of expressive features such as wrinkles both temporally and spatially is relatively unexplored. We demonstrate a motion-capture process that needs only the addition of synchronized video cameras to capture expression wrinkles.

We have implemented a prototype that demonstrates our approach, and we

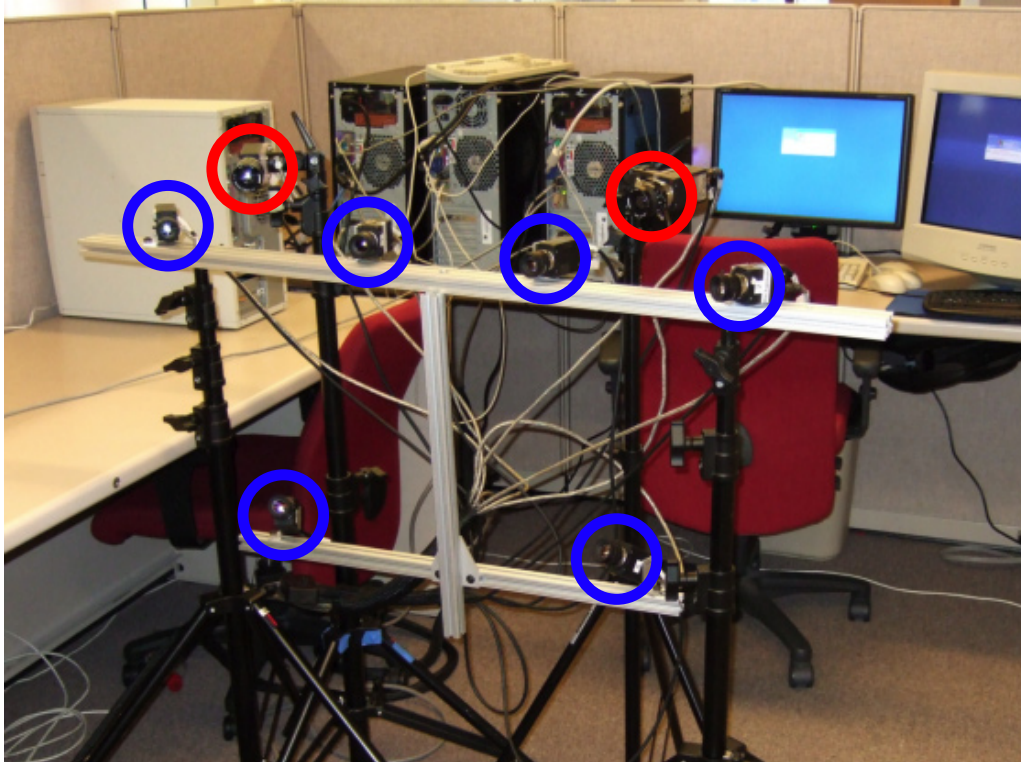


Figure 4.1: Face-capturing setup. *Our system consists of six cameras (indicated by blue circles) for tracking large-scale facial motions based on 80–90 marker points, complemented by two cameras (red circles) for detecting and fitting medium-scale expression wrinkles.*

show results for performance replay and wrinkle processing in Section 4.4. Our method creates high-quality facial animations without the intervention of a skilled artist.

4.2 Data Acquisition

In this section we describe the data acquisition process for capturing the different components of our face model at various spatial and temporal resolutions.

The **static, high-resolution face mesh** is acquired using a commercial face-scanning system from 3QTech ¹. We use the method of [Nehab et al., 2005]

¹www.3qmd.com

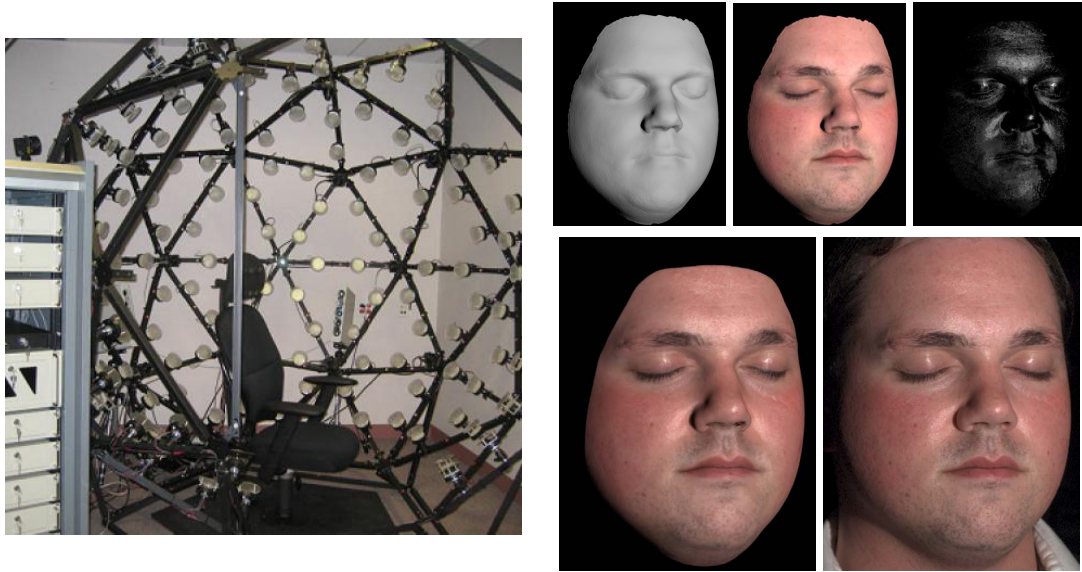


Figure 4.2: High-resolution geometry and appearance data. *Left: Light dome combined with a commercial face-scanning system from 3QTech [Weyrich et al., 2006]. Right, upper row: Rendering of individual components of the output data, consisting of a static, high-resolution face mesh, albedo, and coefficients of the spatially varying Torrance-Sparrow BRDF model. Right, lower row: Side-by-side comparison of a rendered model and real photograph under similar lighting conditions.*

to improve the acquired geometry using photometric stereo, thereby successfully capturing even fine-scale geometric details. The acquisition process takes 30 seconds and produces a face mesh F with approximately 500k–700k vertices, depending on the face size. We also acquire reflectance data and compute the albedo texture, spatially-varying coefficients of the Torrance-Sparrow BRDF model, and subsurface scattering parameters [Weyrich et al., 2006]. Figure 4.2 illustrates the acquisition setup and the the individual components of the appearance model.

The **faster, large-scale face motion** is captured with a setup consisting of six Basler cameras running at 50 fps with a resolution of 656×490 pixels (Figure 4.1). The cameras are placed slightly staggered so that each point of the face is clearly visible in at least two cameras. All cameras are synchronized using an external trigger signal from a USB I/O device². We track about 80–

²<http://www.datx.com/econ>

4 Performance Capture

90 marker points on the face, which are painted blue to maximize the color difference with skin.

The tracking requires a correspondence between points in the video footage, which we establish by consistently labeling them by hand in the first frame of each camera. The labeled 2D points are then tracked throughout the whole sequence independently for each camera. After establishing the intrinsic camera parameters [Svoboda et al., 2005] we use a standard triangulation method to compute the 3D location $\mathbf{m}_{i,t}$ of every marker i in every frame t . To suppress noise in the reconstructed 3D positions, we apply a spatio-temporal bilateral filter to the marker positions, which reduces smoothing for time frames with large movement. This controlled smoothing is important for preserving convincing facial expressions.

To capture the **slower, medium-scale expression wrinkles** we add two high-resolution Basler cameras with 12.5 fps and 1384×1038 pixels. These cameras run exactly four times slower than the motion-capture cameras, making the synchronization easier. All cameras are extrinsically calibrated so that the reconstructed motion-capture performance can be easily projected into the views of the high-resolution cameras.

The scene is captured under approximate ambient uniform illumination, without any light source intensity calibration. We assume that the subject faces approximately the same direction throughout the acquisition process.

4.2.1 Mocap / Face Correspondence

In order to transfer the mocap displacements $(\mathbf{m}_{i,t} - \mathbf{m}_{i,0})$ to displacements $\mathbf{u}_{i,t} = (\mathbf{f}_{i,t} - \mathbf{f}_{i,0})$ in face space we have to establish a correspondence map between the mocap points and the 3D face mesh as illustrated in Figure 4.3.

For that we pick the mocap frame most similar in facial expression to the face scan. Let us assume without loss of generality that this is the first frame, consisting of the points $\mathbf{m}_{i,0}$.

The user first manually selects the corresponding vertex positions $\mathbf{f}_{i,0} \in F$ by clicking on the face mesh. Given this coarse set of corresponding points, position, orientation, and scaling of the face mesh could in principle be adjusted using Horn’s shape matching method [Horn, 1987]. Since the mocap points and the face mesh were captured from the same person, the resulting rigid registration would be quite accurate. However, subtle variations in facial expression, e.g., in the opening angle of the mouth, would not be accounted for.

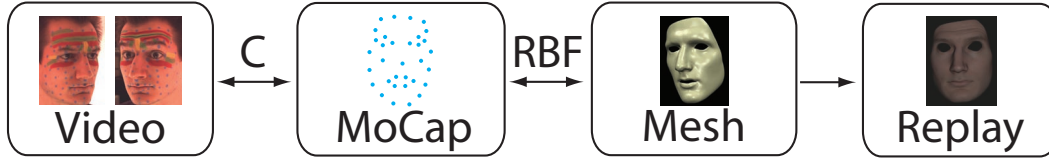


Figure 4.3: Establishing correspondence. Camera calibration C allows to reconstruct 3D positions of markers. Although the initial frame of the mocap points and the face mesh were captured from the same person in neutral pose, subtle variations in facial expression could occur. We therefore employ Radial Basis Functions (RBF) to achieve a smooth correspondence space warp.

Therefore, we use a non-rigid registration technique that interpolates the discrete point correspondences over space in order to achieve a smooth correspondence space warp $\mathbf{c} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. Similar to [Noh and Neumann, 2001], we use radial basis functions (RBFs) for this scattered data interpolation problem, which represents the function \mathbf{c} as

$$\mathbf{c}(\mathbf{x}) = \sum_{i=1}^n \mathbf{w}_i \cdot \phi(\|\mathbf{x} - \mathbf{c}_i\|) + \mathbf{q}(\mathbf{x}) , \quad (4.1)$$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a scalar basis function, $\mathbf{w}_i, \mathbf{c}_i \in \mathbb{R}^3$ are the weights and centers of the RBF, and $\mathbf{q} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a quadratic trivariate polynomial. In order to find the RBF that interpolates the constraints, i.e.,

$$\mathbf{c}(\mathbf{m}_{i,0}) = \mathbf{f}_{i,0}, \quad i = 1, \dots, n ,$$

the centers are chosen to coincide with the constraints, i.e., $\mathbf{c}_i = \mathbf{m}_{i,0}$. This results in a symmetric linear system to be solved for the weights \mathbf{w}_i and the coefficients of the quadratic polynomial \mathbf{q} [Carr et al., 2001]. In contrast to [Noh and Neumann, 2001] we use the triharmonic RBF basis function $\phi(r) = r^3$, which yields a smooth C^2 function of provable global fairness [Duchon, 1977, Botsch and Kobbelt, 2005]. Although the resulting linear system is dense, it can be solved efficiently since the number of constraints n is < 100 . Notice that because of the polynomial term $\mathbf{q}(\mathbf{x})$, the function \mathbf{c} can exactly reproduce affine motions, which makes a rigid pre-registration unnecessary.

Given the space warp \mathbf{c} , we now have to transform the mocap displacements $(\mathbf{m}_{i,t} - \mathbf{m}_{i,0})$ into face space. For a similar setting, [Noh and Neumann, 2001] proposed a heuristic to transfer displacement vectors from one *mesh* onto another by adjusting the displacements' scaling and orientation based on local

4 Performance Capture

frames and local bounding boxes associated with mesh vertices. In contrast, we want to transform displacements from only a coarse point cloud $\mathbf{m}_{i,0}$ to a face mesh, and hence cannot use their surface-to-surface heuristic.

However, the space warp \mathbf{c} already contains all the required information to transfer the mocap displacements: We simply use \mathbf{c} to transfer the displaced mocap points $\mathbf{m}_{i,t}$, which yields $\mathbf{f}_{i,t}$. From those points we compute the face-space displacements as

$$\mathbf{u}_{i,t} = \mathbf{c}(\mathbf{m}_{i,t}) - \mathbf{f}_{i,0} .$$

4.3 Medium-Scale Feature Reconstruction

In this section, we first describe an image-based algorithm for tracking wrinkles in video data, fitting 2D B-splines to them, and estimating their cross-section shapes from self-shadowing effects. Then, we describe a physically-inspired *non-linear* shell deformation model that, with the 2D data as input, allows us to synthesize *medium-scale* 3D expression wrinkles and bulging onto the large-scale animation.

4.3.1 Wrinkle Tracking

In the spirit of shape-from-shading, we exploit self-shadowing effects to track wrinkles and estimate their properties. In the acquisition process, wrinkles are marked with a diffuse color, as shown in Figure 4.4. It masks the underlying skin, making the depth estimation more robust and independent of skin type and pigmentation, e.g., freckles. Furthermore, to simplify the tracking we choose colors that are clearly silhouetted against skin albedo. Neighboring wrinkles that are close to each other are marked with different colors. Our lighting setup produces approximately uniform ambient illumination.

The first step in wrinkle tracking is to find image pixels associated with each predefined wrinkle. We use a binary support vector machine (SVM) with L2 soft margin and RBF kernel [Cortes and Vapnik, 1995] to classify the video images into wrinkle and non-wrinkle patches. It turned out that training the machine was easy. In most cases it was sufficient to create a binary mask for the first image in the video, and use this as training data for estimating the support vectors that were then used for classifying the remaining video images. In case of multiple wrinkles and thus multiple marker colors, the

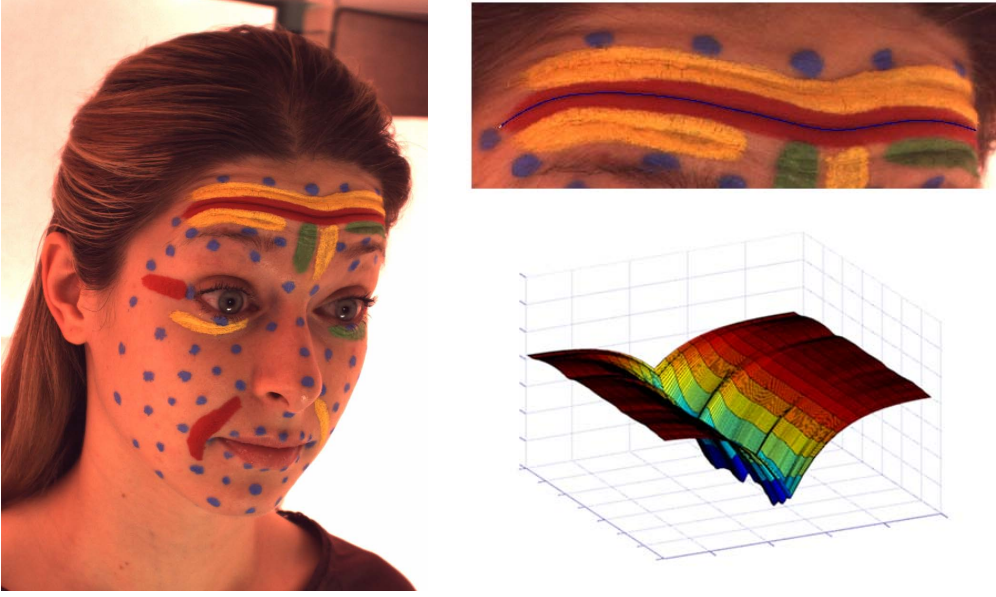


Figure 4.4: For each wrinkle marked in the video (left) a B-spline curve $\mathbf{v}(x)$ is fitted (top right) and corresponding cross-section shapes $S(w, d, p)$ are extracted (bottom right).

binary support vector machine is trained and applied for each color independently. We apply morphological operations (e.g., erosion and dilation) to remove possible pixel classification errors caused by noise.

For wrinkle patches, we represent each wrinkle valley in a compact and smooth way using a uniform B-Spline curve $\mathbf{v} : \mathbb{R} \rightarrow \mathbb{R}^2$. For each patch of wrinkle pixels $\{\mathbf{p}_1, \dots, \mathbf{p}_k\}$, we perform a PCA, resulting in a mean pixel position $\bar{\mathbf{p}}$ and the patch's principal axis \mathbf{a} . We parameterize the pixels \mathbf{p}_i by their position x_i along the axis \mathbf{a} , i.e.,

$$x_i := x(\mathbf{p}_i) = (\mathbf{p}_i - \bar{\mathbf{p}})^T \mathbf{a}.$$

Since wrinkles do not deviate too much from straight lines, this kind of parameterization does not cause any problems. The number of control points is chosen between 5–12, depending on the length of the wrinkle.

The spline $\mathbf{v}(x)$ is fitted in a weighted least-squares sense, minimizing an energy

$$E_{\text{spline}} = \sum_{i=1}^k w_i \|\mathbf{p}_i - \mathbf{v}(x_i)\|^2 \quad (4.2)$$

that measures the Euclidean distance from the pixels \mathbf{p}_i to the valley curve $\mathbf{v}(x)$. We weight each pixel \mathbf{p}_i with a value w_i inversely proportional to

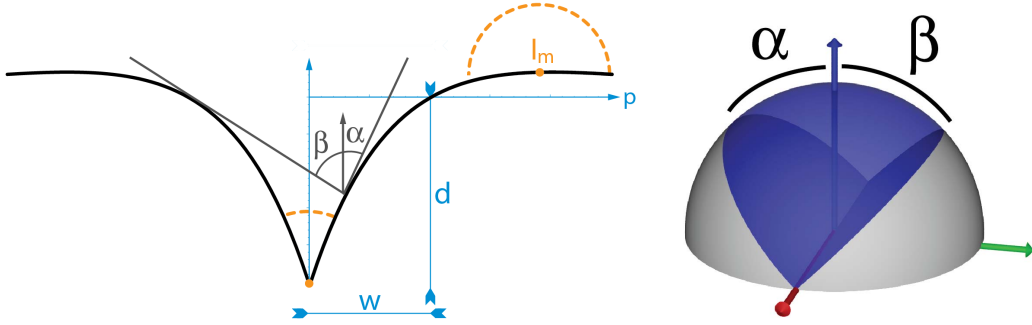


Figure 4.5: *Left: Wrinkle cross-section function $S(w, d, p)$. At point m the observed intensity $I_{\text{obs}}(m)$ is maximal because no incoming light is blocked, in contrast to points $p \in [-m, m]$. Right: The observed intensity $I_{\text{obs}}(p)$ at an arbitrary point p of the wrinkle is computed by integrating the incoming light over the hemisphere. The two angles α and β determine the area of the spherical wedge of incoming (blue) and blocked (gray) light.*

its gray-scale intensity g_i , $w_i = (g_{\max} - g_{\min}) / (g_i - g_{\min})$, where g_{\max} and g_{\min} are the maximum and minimum of the observed gray-scale intensities in the wrinkle segment. Due to self-shadowing effects, wrinkle valleys are darker than their surroundings, and our weighting strategy favors the B-spline curve that follows the wrinkle valley.

4.3.2 Cross-Section Shape Estimation

A wrinkle cross-section can be classified into two characteristic parts: the wrinkle valley and the bulges, one on each side. We have designed a method for locally estimating the gradient of wrinkle valleys from the ratio of observed image intensity on the bulges and the valley. Our method exploits the self-shadowing effect of wrinkles, assuming a Lambertian surface (thanks to the use of a diffuse marker color) and uniform but unknown ambient illumination. We define wrinkle gradient through width $w(x)$ and depth $d(x)$ in image space, varying along the parameterization x of each valley spline $\mathbf{v}(x)$. Later, in 3.3.1, we describe the projection of image space wrinkles onto the large-scale facial animation. Following Bando et al. [Bando et al., 2002], we use an analytic function to model the cross-section of a wrinkle valley. The wrinkle bulge is a more complex phenomenon involving neighboring wrinkles and, unlike Bando et al., we model it separately as described in the next

4.3 Medium-Scale Feature Reconstruction

subsection. With p the distance orthogonal to the spline $\mathbf{v}(x)$, the wrinkle cross-section at x is modeled by the function

$$S(p) = S(w, d, p) = d \cdot \left(\frac{p}{w} - 1 \right) \cdot e^{-p/w}.$$

Then, the intensity at a point $(p, S(p))$ on this cross-section (under ambient illumination I_{ambient}) can be locally estimated by employing a 2.5D model (Figure 4.5) to integrate the incoming light over a hemisphere Ω :

$$I(p) = \frac{1}{\pi} \int_{\Omega} V(p, \omega) \cdot \left(\mathbf{n}(p)^T \omega \right) \cdot I_{\text{ambient}} \, d\omega, \quad (4.3)$$

where $\mathbf{n}(p)$ is the normal vector to the curve $S(p)$ and $V(p, \omega)$ is the visibility function, which is 1 if $(p, S(p))$ is visible from direction ω and 0 otherwise. Notice that if no incoming light is blocked, i.e., $V(m, \omega) = 1, \forall \omega$, the intensity is maximum, $I_{\text{max}}(p) = I_{\text{ambient}}$.

For a given wrinkle shape $S(p) = S(w, d, p)$, the visibility function $V(p, \omega)$ can be computed from the apex angles α and β of the spherical wedge (Figure 4.5), representing all directions of incoming light. These angles are given by the tangent at point $(p, S(p))$ of the wrinkle shape and the tangent at the opposite valley shape going through $(p, S(p))$. Once α and β are computed, the hemisphere integral Equation (4.3) turns into a 2.5D integral over the visible spherical wedge

$$I(p) = \frac{1}{\pi} \int_{-\alpha}^{\beta} \int_0^{\pi} \left(\mathbf{n}(p)^T \omega \right) \cdot I_{\text{ambient}} \, d\omega.$$

Our goal is to find the cross-section parameters d and w such that the computed intensities $I(p)$ match the intensities $I_{\text{obs}}(p)$ observed in the image.

Assuming that there is a point m on the wrinkle bulge without self-shadowing, we can estimate the ambient illumination, $I_{\text{ambient}} \approx I_{\text{obs}}(m)$. Then, we can work with the ratio $I(p) / I(m)$, which is independent of I_{ambient} , and compute the wrinkle-shape parameters without measuring or calibrating the light-source intensity. We obtain the intensity values of a cross-section $I_{\text{obs}}(p)$ by extracting the pixel values perpendicular to the valley spline $\mathbf{v}(x)$ in 2D image space. Then, we compute d and w by minimizing the non-linear least-squares problem (using Matlab's Gauss-Newton optimization)

$$\min_{d,w} \sum_p \left\| \frac{I_{\text{obs}}(p)}{I_{\text{obs}}(m)} - \frac{\int_{\Omega} V(p, \omega) \left(\mathbf{n}(p)^T \omega \right) \, d\omega}{\pi} \right\|^2. \quad (4.4)$$

If no wrinkle is present, the fitted depth d is 0.

4.4 Results

This section presents still images from various animation sequences computed with our model. To see the full model performance please see the accompanying video. All images and animations in this paper were rendered using an extended version of PBRT³ that supports skin subsurface scattering. The facial reflectance data as well as the high-resolution facial geometry were acquired using the hardware described in [Weyrich et al., 2006].

4.4.1 Performance

In this section we give timings for the different stages of processing the video data and animating the face mesh. Since the processing times are almost equal for the different subjects we list only average timings. All computations were carried out on a standard PC with an Intel Pentium 2.8 GHz and 1 GByte of main memory.

The large-scale linear animation involves the computation of the correspondence RBF (Equation 4.1) and the solution of the bi-Laplacian linear system (Equation 3.3) for the actual surface deformation. The RBF interpolation can be solved within milliseconds due to its small size. After a pre-factorization of about 120s, the surface deformation can be performed at a rate of about 3s per frame.

For each video frame, the wrinkle-capture process takes about 5s for image segmentation and spline fitting, and about 8min for the non-linear cross-section estimation, which currently is implemented in Matlab. The medium-scale wrinkle synthesis projects the extracted 2D wrinkles onto the large-scale animation and solves a non-linear minimization of stretching and bending, which is the dominant cost of about 20min. The final rendering takes about 10min per frame in high quality mode.

4.4.2 Expression Replay and Wrinkle Editing

Figure 4.6 depicts a sequence of still images with varying facial expressions for two different subjects. The images were taken from the video animation and show replays of facial expressions animated with our model. For all facial animations, we cut out the subjects' eyes, and the meshes were clipped along the hair and ear lines. In the second column from left, we show the

³<http://www.pbrt.org/>

deformed facial geometry as computed by our large-scale linear deformation model. Note that at this stage, the faces do not contain any expression wrinkles. The third (geometry only) and fourth (skin rendering) columns show the results after adding wrinkles to the deformed model. The facial expression of the female subject in the upper row has large forehead wrinkles that are modeled and animated very realistically by our model. The performance of the male subject primarily leads to wrinkle formations around the eyes, and our model captures the resulting deformations very convincingly.

Figure 4.7 presents two standard facial expressions, “astonished” and “angry,” which lead to different wrinkle formation. Despite the lack of self-collision detection, our model replays these deformations very well. An illustration of our editing capabilities is given in Figure 4.8. In this sequence, we gradually scaled the wrinkle depth to weaken or enhance the effect of the forehead wrinkles. The skin bulges created by our wrinkle model provide a realistic deformation of the facial skin in all images of this sequence. The figure also shows the flexibility of our multi-scale model on (per-frame) manual edits. The rightmost image shows a single frame edit, where the nasolabial wrinkles were added manually simply by drawing their valley splines into the video frame and specifying depth and width parameters.

4.5 Discussion

In this chapter we presented a performance capture system for three-dimensional dynamic face models that acquires the different types of spatial and motion scales. By simply combining two video cameras with an optical marker-based motion capture system, we were able to efficiently estimate all required coefficients of our model for facial geometry and motion presented in Chapter 3.

The acquisition and hence the ultimate quality is currently limited by the frame rate of the cameras and by the homemade motion tracker we utilized to produce our results. However, this is not an inherent limitation of the model, because it could easily be alleviated by taking commercially available high-speed cameras and motion-tracking systems.

An important extension of our approach is an explicit representation of small wrinkles. While our representation should scale directly to small features as well, the finite camera resolution and the explicit coloring of wrinkles artificially limits their actual minimum size. Therefore, it would be interesting to extend our multi-scale model to explicitly represent and animate small-scale wrinkles. Recently, [Beeler et al., 2010] showed a passive stereo system

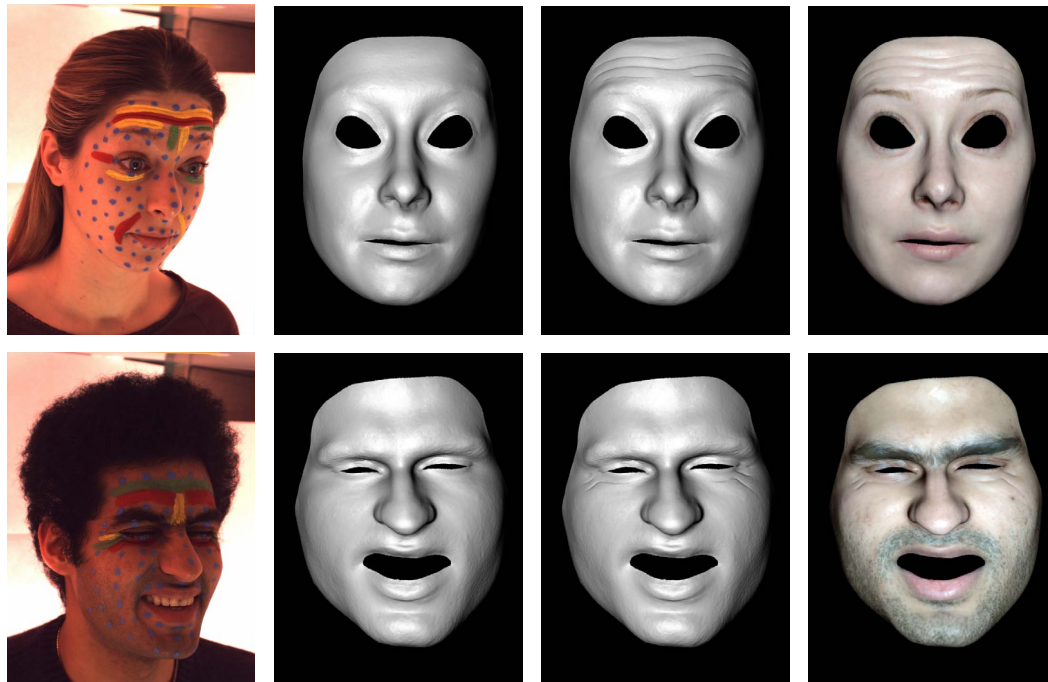


Figure 4.6: Performance replay of captured video sequences (left) of two different subjects. The large-scale linear animation first deforms the high-resolution face mesh based on tracked mocap markers (center left). The missing medium-scale expression wrinkles are synthesized by a non-linear energy minimization (center right). The rightmost column shows high-quality skin rendering including subsurface scattering.



Figure 4.7: Two more examples showing facial expressions for the standard emotions "astonished" (left) and "angry" (right).



Figure 4.8: *Our multi-scale face model enables wrinkle processing by scaling the depth parameters extracted from video. This allows us to either weaken (50%) or enhance (200%) the original wrinkles (100%). The rightmost image shows a single frame edit, where the nasolabial wrinkles were added manually simply by drawing their valley splines into the video frame and specifying depth and width parameters.*

for capturing the 3D geometry of a face in a single-shot under standard light sources with sub-millimeter accuracy. In theory, their shape-from-shading approach for estimating small-scale details could be combined with our representation.

Finally, we are interested in wrinkle animation transfer between individuals, an issue of high practical relevance for applications in the special-effects industry. Chapter 5 will address this issue, and will show how this model can be applied and extended for efficient animation and transfer of facial details.

Real-time Animation and Transfer of Facial Details

In the previous Chapters 3 and 4 we presented methods to acquire and model detailed expressive facial deformations. However, facial deformations are computed in an off-line process, which excludes its applicability from real-time applications such as video games, interaction with virtual avatars, or real-time editing by animators. In general, current approaches are either computationally expensive, require very specialized capture hardware, or are extremely labor intensive. In this chapter, we present a method for real-time animation of highly-detailed facial expressions based on sparse motion capture data and a limited set of static example poses. It naturally extends our representation and model for facial geometry and motion (Chapters 3), and can be driven by its model parameters. The proposed in-game algorithm is fast, easy to implement, and maps well onto programmable GPUs. Furthermore, we demonstrate in this chapter that this representation allows to easily transfer facial details onto existing 3D face models and animations that lack of those, thereby increasing the visual expressiveness.

5.1 Motivation

Data-driven approaches avoid complex computations by blending example poses from a database of facial expressions [Pighin et al., 1999, Blanz and Vetter, 1999, Vlastic et al., 2005]. However, spanning the space of expressive motions requires a large and complex-to-acquire database [Fordham, 2003], and deriving the blending weights becomes difficult and/or expensive [Joshi et al., 2003]. Facilitated by the steadily increasing CPU performance, *computational approaches* are nowadays able to realistically model facial details using complex non-linear deformation models [Sifakis et al., 2005, Bickel et al., 2007], but are still tied to high computational cost. As shown in Chapter 3, facial motion can be decomposed into large-scale global motion and fine-scale details, such as expression wrinkles. While the first can efficiently be computed using a linear shell deformation model, their fine-scale wrinkle synthesis still requires a computationally expensive non-linear thin-shell model.

In this chapter, we propose a *hybrid animation* technique that combines computational and data-driven approaches, and thereby achieves detailed facial expressions that are intuitive to control as well as efficient to compute. Building on the decomposition of our model into different scales (Chapter 3), we compute the large-scale motion using the same linear shell deformation, but in contrast employ a data-driven approach for learning fine-scale facial details from a small set of example poses.

While skin wrinkling is a highly non-linear effect, there is a clear relationship to stretching and compression of the underlying skin. We therefore propose a novel pose-space deformation (PSD) technique, which in a preprocess learns the correlation of wrinkle formation to sparsely measured skin strain. At run-time, given an arbitrary facial expression, it computes the skin strain and derives from it the necessary fine-scale corrections for the large-scale deformation.

This hybrid animation technique has several advantages over purely computational or purely data-driven approaches. First, our technique is very accurate: The large-scale deformation is driven by a sparse set of positional constraints, such as motion-capture (*mocap*) markers or user-defined handle points, which are interpolated exactly and therefore enable precise control. The relation of skin strain to fine-scale wrinkles can be learned more efficiently than the relation of marker positions to a non-linearly deformed face. As a consequence, our pose-space deformation works very well even with just a few example poses, although arbitrary accuracy can be obtained by adding more poses.

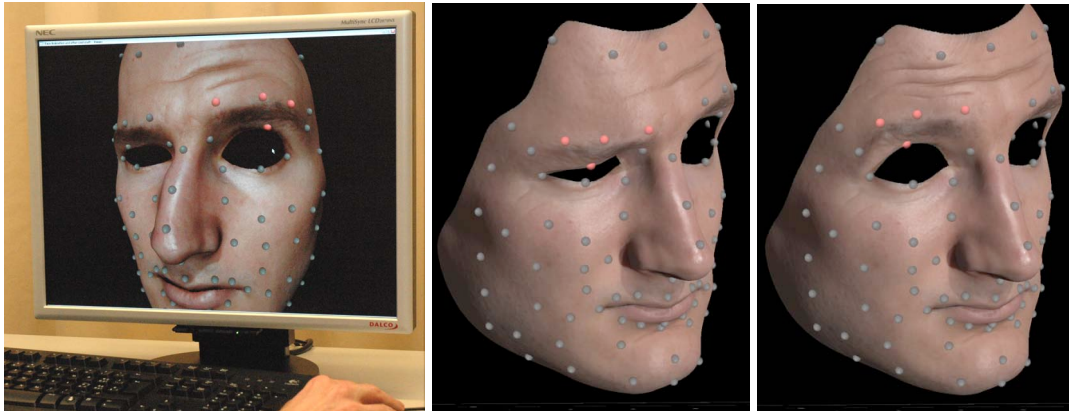


Figure 5.1: *Interactive Editing of Highly-Detailed Faces.* The user intuitively edits a 1M-triangles face model by dragging the handle points near the eyebrow. The face deforms in real-time and produces realistic fine-scale wrinkles.

Our hybrid approach allows large-scale and fine-scale deformations to be solved as superposition of precomputed response functions. Exploiting this, we derive a highly parallel GPU implementation that achieves real-time animations of detailed facial expressions. Those animations are controlled through a set of marker positions, which in combination with real-time performance yields an intuitive editing system for realistic face animations (cf. Figure 5.1).

Face models in games, virtual reality, or computer animation often appear unrealistic due to the lack of facial details. Once learned from a set of example poses, our explicit pose-space representation of fine-scale details allows us to easily transfer them onto other animated faces, thereby enhancing their visual plausibility without changing their overall motion. As such, our method provides a simple way to re-use captured or manually modeled high-res face data.

5.2 Overview

We start with an overview of the input data and the workflow of our hybrid face animation pipeline, depicted in Figure 5.2, before describing the large-scale and fine-scale deformations in more detail in the following sections.

In contrast to blend shapes, which linearly combine several facial poses, we use a single high-resolution **face mesh** \mathcal{F} of the rest pose ($\approx 1\text{M}$ triangles),

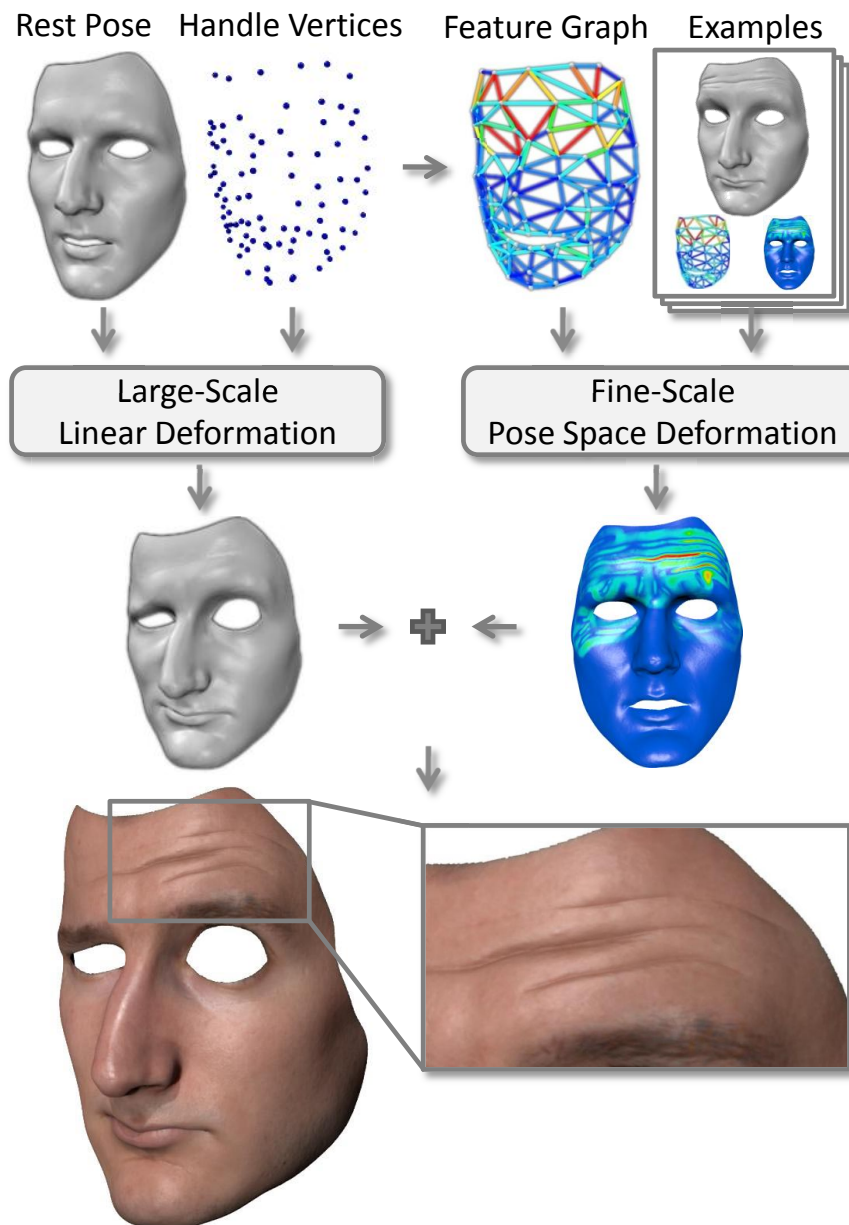


Figure 5.2: Our hybrid face animation pipeline computes the large-scale facial motion from a linear deformation model, and adds fine-scale details using a pose-space deformation that learns skin corrections from a set of example poses.

and generate all frames of the animation by deforming \mathcal{F} from its rest pose. Following our model for facial geometry and motion (Chapter 3), we compute the large-scale facial motion using a linear thin shell model (Section 5.3). Its deformation is controlled through a set of $H \approx 100$ **handle vertices**, which may correspond to a set of face points tracked during an actor’s performance, or may be interactively controlled by an animator. The resulting large-scale deformation successfully captures the overall facial expression, but lacks fine-scale facial details such as expression wrinkles (Figure 5.2, left).

The non-linear behavior that goes beyond the linear large-scale motion is learned in a preprocess from a set of P **example poses**. These examples are represented by high-resolution triangle meshes in full vertex correspondence with the rest pose \mathcal{F} . In practice, example poses can for instance be created manually by an artist, by capturing an actor’s face with a high-resolution scanner [Zhang et al., 2004, Ma et al., 2007], or by reconstructing a detailed face mesh with expression wrinkles as demonstrated in the previous chapter. Given the example poses, the corresponding fine-scale details are extracted as the difference between the examples and the results of the large-scale deformation for the same poses, and are stored per-vertex in local tangent frames.

In order to interpolate the fine-scale corrections of the examples onto other poses, we define a novel pose space for facial expressions based on skin strain, and learn the formation of skin bulges and wrinkles in this space. To this end, we construct a **feature graph** from the H handle vertices, following approximately the contour of the face, but without the need to produce a consistent mesh (e.g., edges may cross each other). The feature graph’s edges constitute the locations where skin strain is measured (Section 5.4.1). For each new pose in an animation, we compute the skin strain according to the deformed feature graph, and use this information for a pose-space correction of fine-scale facial details (Figure 5.2, right, Sections 5.4.2 and 5.4.3).

5.3 Large-Scale Deformation

For the large-scale face deformations we employ the linear deformation model described in Section 3.2.1. This section describes how the deformation can be computed in real-time, even for high-resolution meshes. Our model is based on the assumption that displacements of a set of sparse handles provide sufficient geometric constraints for capturing the large-scale facial motion in a plausible way. Given as input constraints the 3D displacements $\mathbf{u}_H \in \mathbb{R}^{H \times 3}$ of the H handle vertices, the large-scale deformation \mathbf{u} is com-

5 Real-time Animation and Transfer of Facial Details

puted by minimizing a simplified, quadratic thin shell energy. This amounts to solving the corresponding Euler-Lagrange equations

$$-k_s \Delta \mathbf{u} + k_b \Delta^2 \mathbf{u} = 0, \quad (5.1)$$

under the constraints \mathbf{u}_H imposed by handle vertices. In this equation, k_s and k_b denote the stiffness for surface stretching and bending, respectively, and Δ is the Laplace-Beltrami operator. Discretizing the latter using the cotangent weights [Meyer et al., 2003] yields the linear system

$$\begin{pmatrix} A & A_H \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{u}_H \end{pmatrix} = \mathbf{0},$$

to be solved for the unknown displacements $\mathbf{u} \in \mathbb{R}^{(V-H) \times 3}$ of the $(V - H)$ free vertices.

In contrast to solving this system for each frame using a sparse Cholesky factorization [Toledo et al.,], we precompute the “basis function” matrix $B = -A^{-1}A_H$, which depends only on the rest-pose mesh \mathcal{F} and the selection of handle vertices. After factorizing A once, each column of B is computed by solving a sparse linear system involving A and the corresponding column of A_H . At run-time, the large-scale deformation \mathbf{u} is obtained from the handle displacement \mathbf{u}_H by the matrix product $\mathbf{u} = B \cdot \mathbf{u}_H$, which can efficiently be computed in a parallel GPU implementation.

5.4 Fine-Scale Deformation in Pose Space

The linear deformation model described in the last section approximates the facial motion well at a large-scale. However, non-linear fine-scale effects, such as the bulging produced by expression wrinkles, cannot be reproduced. These fine-scale deformations are highly non-linear with respect to the positions of the handle vertices. However, they vary smoothly as a function of facial pose, hence we have opted for learning, as a preprocess, the fine-scale displacement \mathbf{d} from a set of example poses, and then at run-time compute it by interpolation in a suitable facial pose space.

We first define a facial pose space based on a rotation-invariant feature vector of skin strain (Section 5.4.1), and formulate the learning problem as scattered-data interpolation in this pose-space (Section 5.4.2). We extend the basic method to *weighted* pose-space deformation (Section 5.4.3), thereby allowing for a more compact basis. Based on our explicit pose-space representation we describe the transfer of fine-scale details onto novel faces and animations (Section 5.4.4).

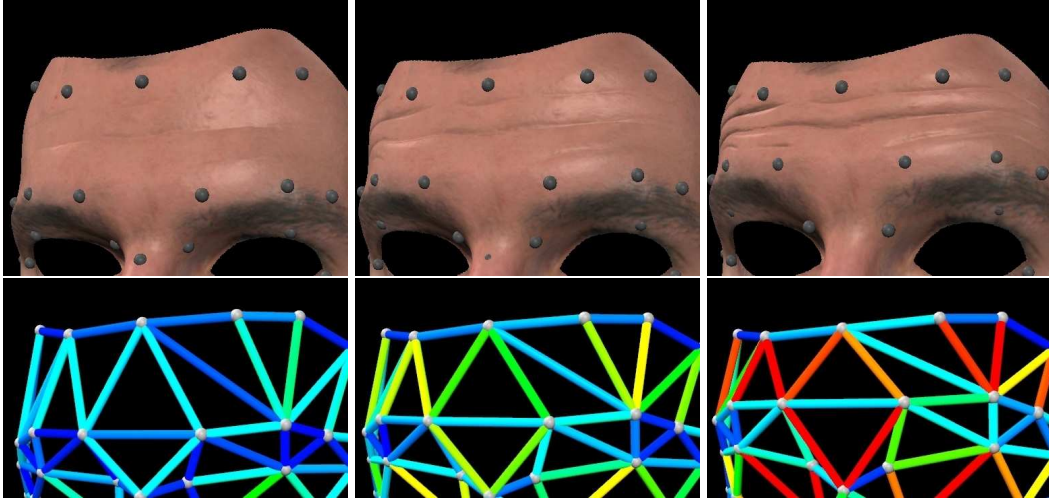


Figure 5.3: *Wrinkle Formation and Skin Strain. Close-up of the forehead for three input examples (top) and the corresponding strain on the feature edges (bottom), showing the correlation of lateral skin strain and wrinkle formation.*

5.4.1 Definition of the Feature Vector

At each animation frame, the raw input data describing a pose consists of the positions of the handle vertices. This data is not invariant under rigid transformations, hence it does not constitute an effective pose descriptor, which, however, is required for the pose-space deformation method described in the next section. Bulging effects of wrinkles appear due to lateral compression of skin patches [Wu et al., 1996]. Exploiting this correlation, we suggest a feature vector that measures skin strain at various points across the face.

Based on the feature graph connecting the handle points using F edges (Figure 5.2), we define the F -dimensional feature vector $\mathbf{f} = [f_1, \dots, f_F]^T$ of a pose containing in f_i the relative stretch of the i 'th feature edge, which can be regarded as a measure of strain. Specifically, given the positions of the endpoints $\mathbf{p}_{i,1}$ and $\mathbf{p}_{i,2}$, and the rest length l_i , we define

$$f_i = (\|\mathbf{p}_{i,1} - \mathbf{p}_{i,2}\| - l_i) / l_i.$$

This feature vector is invariant under rigid body transformations, and the feature distance $\|\mathbf{f} - \mathbf{f}_j\|$ corresponds to the Euclidean norm of relative edge stretch from the j 'th example pose to the input pose. Figure 5.3 shows the correlation between the feature vector and wrinkle formation. Alternatively, a per-vertex strain tensor could be used, but in practice our discrete strain approximation turned out to be sufficient.

5.4.2 Pose-Space Deformation

In order to exploit the connection of skin strain to wrinkle formation we want to learn their functional relation in a pose-space deformation (PSD) framework. To this end, we represent each facial expression by its rotation-invariant feature vector \mathbf{f} as described in the last section. Hence, each facial expression corresponds to a point in an F -dimensional pose space, which constitutes the domain of the function we want to learn. Its range is the fine-scale detail correction \mathbf{d} , represented as well in a rotation-invariant manner by storing them in per-vertex local frames.

Each of the P example poses corresponds to a feature vector \mathbf{f}_i with its associated fine-scale displacement \mathbf{d}_i . Then, PSD corresponds to a scattered data interpolation problem, for which we employ radial basis functions (RBFs). Hence, the function $\mathbf{d} : \mathbb{R}^F \rightarrow \mathbb{R}^{3V}$, mapping a facial pose to 3D fine-scale displacement, has the form

$$\mathbf{d}(\mathbf{f}) = \sum_{j=1}^P \mathbf{w}_j \cdot \varphi(\|\mathbf{f} - \mathbf{f}_j\|), \quad (5.2)$$

where $\varphi(\cdot)$ is a scalar basis function, and $\mathbf{w}_j \in \mathbb{R}^{3V}$ and \mathbf{f}_j are the weight and feature vector for the j 'th example pose. We employ the biharmonic RBF kernel $\varphi(r) = r$, since it allows for smoother interpolation of sparsely scattered example poses than locally supported kernels [Carr et al., 2001].

As a preprocess we compute the RBF weights \mathbf{w}_j in one of two possible ways. If the training dataset of examples consists of only a small number P of extreme poses (e.g., modeled by an artist), we form the basis with all P poses, and compute the weights \mathbf{w}_j such that the displacements of the example poses are interpolated exactly, i.e., $\mathbf{d}(\mathbf{f}_i) = \mathbf{d}_i$. This reduces to solving $3V$ linear $P \times P$ systems, which differ in their right-hand side only.

If the training dataset consists of a large number of example poses with redundant information (e.g., captured from an actor's performance), we select a compact basis of P poses, and compute the weights \mathbf{w}_j that fit all T examples in a least-squares manner. This again amounts to solving $3V$ linear $P \times P$ systems. We select the poses for the basis in a greedy manner [Carr et al., 2001]: Starting with the rest-pose, we incrementally add the pose with largest error and recompute the weights. This pose selection is fully automatic, can achieve arbitrary accuracy, and allows to intuitively specify an approximation threshold.

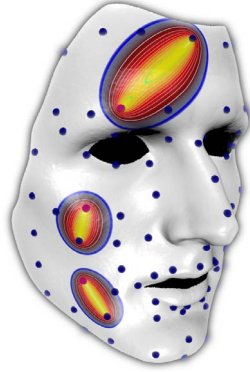


Figure 5.4: *Weight kernel. For three edges, color coded local influence, exponentially dropping from 1 (yellow) to 0 (blue).*

5.4.3 Weighted Pose-Space Deformation

In our basic definition of PSD in Section 5.4.2, every input example influences all vertices of the face mesh in the same manner, since we compute a single feature distance $\|\mathbf{f} - \mathbf{f}_j\|$ per example pose j . As a consequence, the basis of example poses is required to grow exponentially to sufficiently sample the combinations of independent deformations (e.g., raising both eyebrows versus raising just one eyebrow). As an answer to this problem, we adopt a weighted pose-space deformation (WPSD) scheme [Kurihara and Miyata, 2004].

This requires to redefine Equation (5.2) to compute feature distances in a per-vertex manner, replacing the Euclidean metric $\|\mathbf{f} - \mathbf{f}_j\|$ with a weighted distance metric per vertex v :

$$\|\mathbf{f} - \mathbf{f}_j\|_v := \left(\sum_{i=1}^F \alpha_{v,i} (f_i - f_{j,i})^2 \right)^{1/2}, \quad (5.3)$$

where $f_{j,i}$ is the strain of the i 'th feature edge in the j 'th pose. We exploit the fact that the components of our feature vector (i.e., relative stretch of feature edges) measure local properties for assigning weights $\alpha_{v,i}$ based on proximity to the feature edges. Specifically, for a vertex v we define the weight of the i 'th feature edge as

$$\alpha_{v,i} = \frac{\bar{\alpha}_{v,i}}{\sum_i \bar{\alpha}_{v,i}}, \quad \text{with} \quad \bar{\alpha}_{v,i} = e^{-\beta(L_{v,i} - l_i)}, \quad (5.4)$$

where l_i is the rest length of the feature edge, and $L_{v,i}$ is the sum of rest-pose distances from vertex v to the edge endpoints. This weight kernel is 1 on the

5 Real-time Animation and Transfer of Facial Details

edge, and decays smoothly everywhere else, as shown in the adjacent figure. The parameter β can be used to control the degree of decay, based on the local density of handle vertices. In our experiments we discard weights $\bar{\alpha} < 0.025$, and set β such that a vertex is influenced by at most 16 edges.

For WPSD, the computation of RBF weights slightly differs from PSD as described in Section 5.4.2. With function

$$\mathbf{d}_v(\mathbf{f}_i) := \sum_{j=1}^P \mathbf{w}_{v,j} \cdot \varphi\left(\|\mathbf{f} - \mathbf{f}_j\|_v\right), \quad (5.5)$$

the interpolation problem is $\mathbf{d}_v(\mathbf{f}_i) = \mathbf{d}_{v,i}, \forall v \forall i$, i.e., for each vertex v the function \mathbf{d}_v should interpolate v 's displacements in the given example poses. Nevertheless, the weight computation still involves solving $3V$ linear $P \times P$ systems, but the matrix describing the linear system is different for each vertex v . Equivalently to PSD, the WPSD method also yields $3P$ weights to be stored per vertex.

At run-time, given the feature vector \mathbf{f} of the current pose, we evaluate for every vertex the weighted feature distance to each basis pose according to Equation (5.3) and compute the fine-scale displacement vector according to Equation (5.5), accumulating the contributions of the precomputed RBF weights. The computations can be easily parallelized over all vertices, allowing for a highly efficient GPU implementation.

5.4.4 Transfer of Fine-Scale Details

Our representation of fine-scale details furthermore allows to transfer facial details onto new faces and animations. As acquiring or manually modeling high-resolution face geometry is often tedious and expensive, this provides a simple alternative for enhancing otherwise unnaturally smooth looking face models by reusing available fine-scale data.

Due to the fact that the fine-scale deformation is already defined in pose-space, controlled by local skin strain and measured in per-vertex local coordinate frames as described in Section 5.4.2, its transfer reduces to establishing dense correspondence with the target face model.

We establish correspondence by manually marking approx. 40 feature points on the rest-pose of the source face and the target face. After a rigid pre-alignment, the source face is deformed using the large-scale deformation method described in Section 5.3 to approximately fit the target face, and is then projected onto the target face, similar to [Vlasic et al., 2005]. Note that

this is only necessary for a single frame and any other algorithm yielding dense correspondence could be used as well. As source and target model might not have the same mesh topology, we map each vertex of the target mesh to its enclosing triangle on the deformed source mesh using barycentric vertex weights.

Based on the dense correspondence of the source and target rest-pose, the feature graph with its F edges can be mapped in a straightforward manner to the target model. The nodes of the feature graph and the handle vertices on the target model do not have to match. During the animation we first compute the large-scale deformation and hence obtain the necessary feature graph nodes. In case of an existing animation, we do not require the large-scale deformation and instead directly map the feature graph onto the meshes.

The per-vertex RBF weights w_j of the source face model are transferred to the target model by linear interpolation using the barycentric coordinates of the corresponding triangle. The resulting displacements in their local coordinate frame on the target animation are obtained by evaluating Equation (5.5).

5.5 Results and Implementation

This section presents an evaluation and performance analysis of our hybrid face animation method, as well as application experiments, such as animation from mocap data, interactive editing, and transfer of fine-scale details. Please see the accompanying video for the full model performance.

5.5.1 Evaluation and Comparisons

For comparisons, we have used as input 200 frames of an actor’s performance, with high-resolution surface details captured with the method described in the previous chapter (See the top row of Figure 5.7 for some input poses). We use as training dataset the first 100 frames, and the other 100 as ground-truth comparison data. In the training sequence the actor performs roughly 5 expressions. For the comparisons, we implemented two versions of blend shape animation. These methods use a set of blend shapes (key facial expressions) to define a linear space of facial expressions [Parke, 1974].

Global blend shapes use a single weight for each blend shape. We use our feature distance metric explained in Section 5.4.1 to find a convex combination of blend shapes that matches the motion capture marker positions of

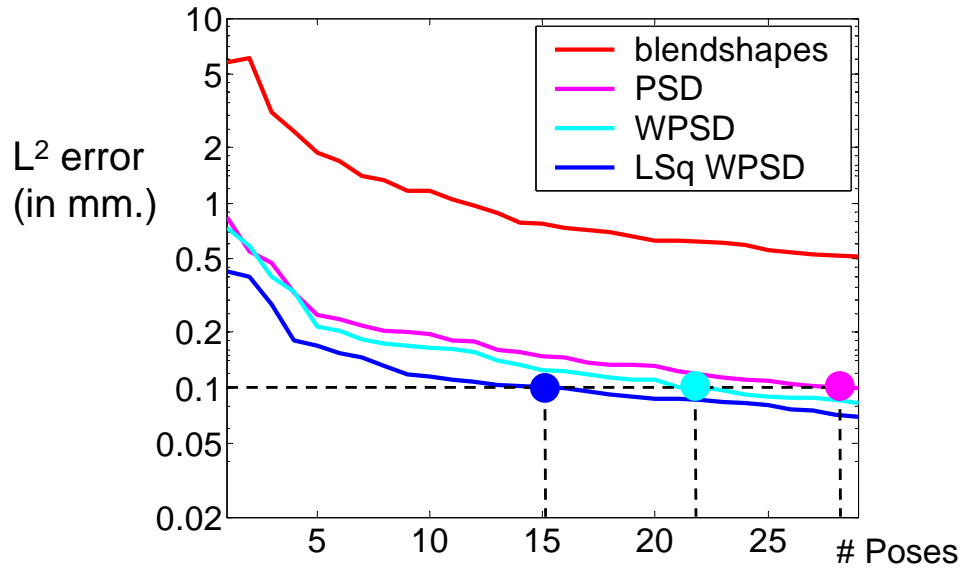


Figure 5.5: L^2 Error Vs. Size of the Basis. Comparison of our implementation of local blend shapes, PSD from Section 5.4.2, and our novel WPSD from Section 5.4.3, both with interpolation of poses and with least-squares fit (LSq), measuring error over a training sequence of 100 frames.

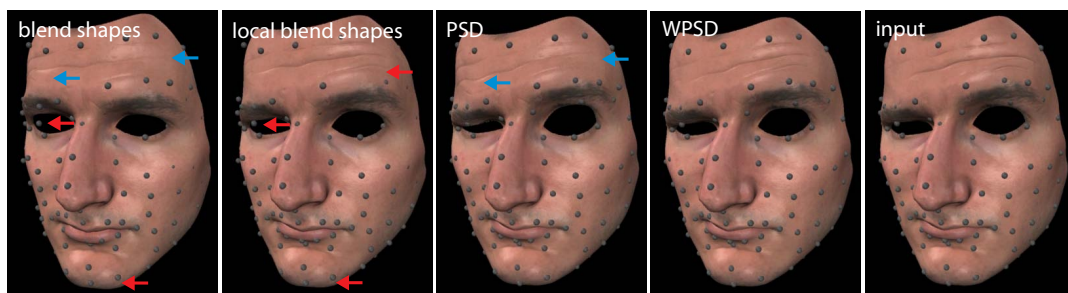


Figure 5.6: Comparison of Methods. Input pose (right) not present in the training data, approximated with various methods from 6 examples (See top row of Figure 5.7). Arrows highlight large-scale (red) and fine-scale (blue) errors.

the actor’s performance. This naive weighting scheme cannot produce expressions outside the convex combination and requires an exhaustive set of blend shapes.

Locally controlled blend shapes typically are based on a segmentation of the face into individual regions, which allows to blend regions separately and thereby alleviates the shortcomings of global blend shapes. We achieve an equivalent behavior using the locally supported, smoothly varying feature distance metric Equation (5.3). This metric allows to locally interpolate blend shapes and can be evaluated in real-time. Other more sophisticated non-real time weight controls are possible [Pighin et al., 1999, Joshi et al., 2003]. However, blend shapes in general suffer from inaccuracy problems as soon as the desired pose is not in the convex space of the example poses.

We compare the blend shape implementation with our **PSD** approach introduced in Section 5.4.2, and our **WPSD** method described in Section 5.4.3, based on both exact interpolation of P poses and on least-squares fitting to T poses.

Figure 5.5 shows the decay of L^2 error for the training sequence T as more examples are added to P . PSD and WPSD achieve a significantly lower error because, in contrast to blend shapes, our face animation method splits facial geometry into large and fine-scale components, guaranteeing that marker positions of the large-scale component are always exactly interpolated.

Figure 5.6 illustrates this for an input pose not present in the training dataset ($P = 6$ poses). Global and local blend shapes suffer from regions of large interpolation error (red arrows). On the other hand, global blend shapes and PSD are affected by incorrect fine-scale details due to lack of local support (blue arrows). Our WPSD model reproduces the input pose best, due to the accuracy of the linear deformation model and local support for synthesizing fine-scale details.

5.5.2 Performance

Compared to the non-linear method in Chapter 3, which requires approximately 20min per frame for synthesizing facial details, we obtain a performance of 4sec/frame using a pure CPU implementation on a similar machine and for the same mesh complexity, which corresponds to a speed-up of a factor of about 300.

GPU Implementation

Moreover, our method allows for efficient parallel GPU-implementation with CUDA [NVIDIA, 2008]. The basis matrix B , RBF weights $\mathbf{w}_{v,j}$, and feature edge weights $\alpha_{v,i}$ are stored in device memory on the GPU. Hence, we need to transfer only the feature vector \mathbf{f} and the displacements of the handle vertices \mathbf{u}_H at run-time. To reduce memory space and access, $\alpha_{v,i}$ is represented as a sparse matrix.

The significant speed up compared to a single core CPU (about a factor of 80, factor of 10 on an eight core) is possible because the method is parallelized over the large number of vertices of the face mesh. Position and displacement of vertices can be computed independently from each other. We need three CUDA kernels to perform the animation. First, a so-called large-scale kernel computes new positions for all face vertices. It performs three dot products for x, y, z of weights with handle positions per vertex. Then, the normals of the face vertices have to be updated. This is done in a kernel with the help of a simple data structure enumerating the one-ring of vertices around every vertex. Then, the fine-scale enrichment kernel calculates the local displacement for each vertex and computes all new vertex positions in parallel. This is followed by a second final vertex normal update for rendering. In our current implementation we utilize the CPU to synchronize the kernel invokes of the before mentioned steps.

Timings

We tested the performance of our method with a detailed facial model for a mesh of $V = 530k$ vertices and more than 1M triangles, with $P = 6$ example poses, $H = 89$ handle vertices, and $F = 243$ feature edges. These settings require storing 137 floats and 24 integers per vertex. We obtain an overall performance of about 30fps on a Nvidia 8800 GTX graphics card, including large-scale deformation (4ms/frame), fine-scale deformation (13ms/frame), two normal vector updates (2×5 ms/frame), and a subsequent simple untextured Gouraud shading rendering in Open GL (4ms/frame). With an additional skin rendering providing real-time subsurface scattering as described in [d'Eon et al., 2007] the final rendering takes about 36ms/frame and we obtain about 15fps. Compared to the non-linear method in Chapter 3, this provides a very significant speed-up of 4 orders of magnitude.

Table 5.1 shows a comparison of performance numbers. Our algorithm is 10 faster than an OpenMP CPU implementation. The CPU implementation would also require an additional copy of the entire face onto the GPU every

	GTX8800 Windows XP	GTX 480 Windows 7	CPU i7 920 QuadCore
Large Scale Deformation	4ms	6ms	64ms
Small Scale Deformation	13ms	11ms	80ms

Table 5.1: Performance numbers for different configurations and comparison to CPU. Face model with 530k vertices. The CPU implementation uses OpenMP to utilize 8 logical cores.

frame before rendering (+2ms GTX480, not included in the measured timings). We also observed that the user space driver model in Windows 7 influences our timings significantly. One solution to reduce the kernel invocation overhead would be to implement everything in a single kernel and use global GPU synchronization techniques. However, those features are only available on newer architectures.

5.5.3 Application Experiments

Animation from Mocap. Given a few poses P , our face animation method allows for high-resolution real-time performance replay from mocap markers, as shown in Figure 5.7. We defined the basis for our example-based fine-scale deformation method using 6 poses from the training dataset as described in the previous section (See top row of Figure 5.7). Then, the same actor performed a different sequence of expressions, captured using only mocap markers, and our method produced a high-quality performance replay in real-time (See side-by-side comparison in the video). The large-scale animation by constrained deformation provides good approximation quality of the overall face, while the fine-scale example-based correction adds high-resolution details.

Interactive Editing. The computational performance and intuitive control by handle vertices allow for interactive editing of face deformations, as shown in Figure 5.1. This might especially be helpful for an artist to fine-tune a mocap sequence. With our pose-space representation of fine-scale details, non-linear wrinkling effects are produced interactively on complex faces simply by dragging the handle points.

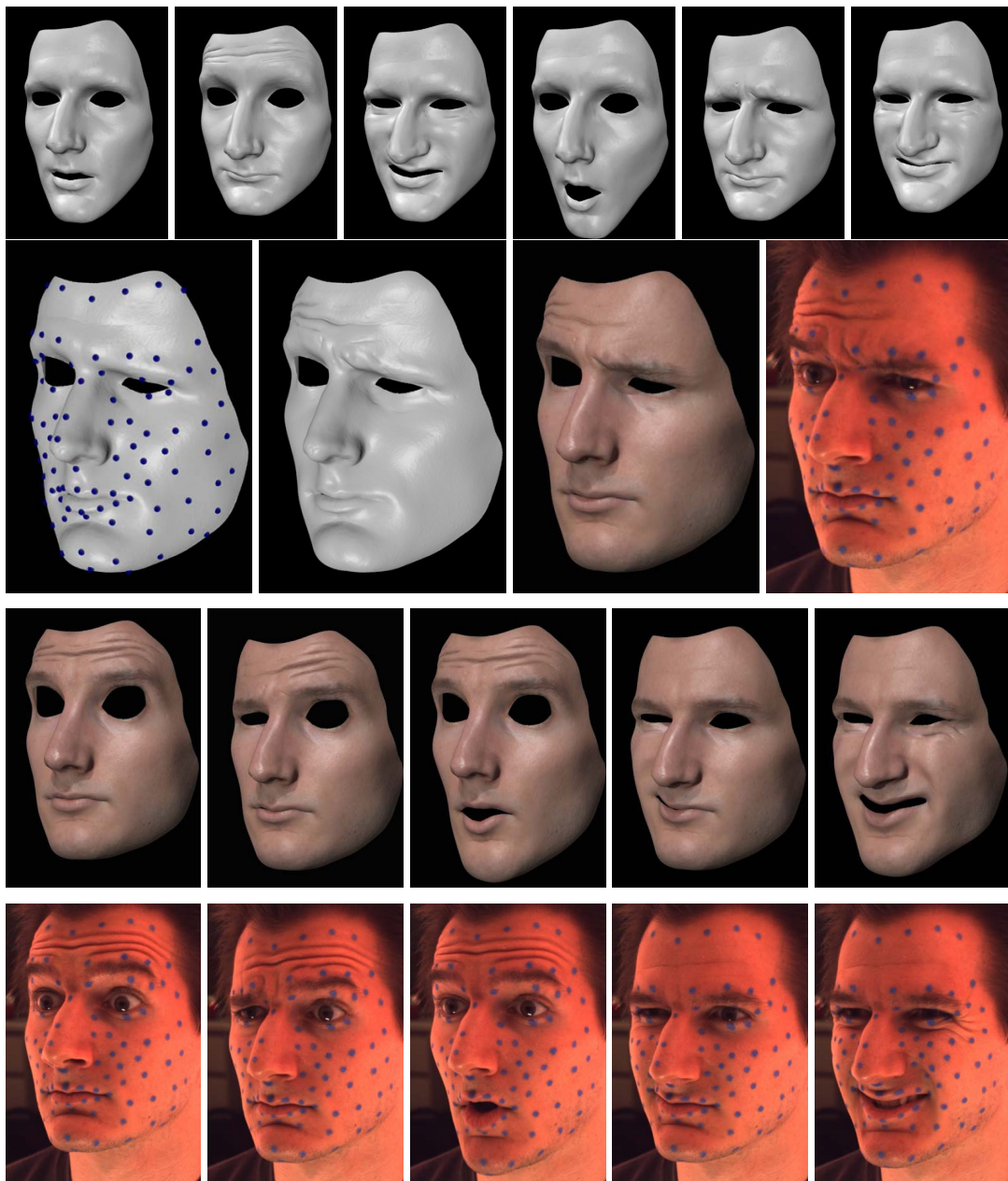


Figure 5.7: *Real-Time Face Animation from Mocap. First row: example poses taken from a different sequence; Second row, from left to right: large-scale deformation interpolating the mocap markers, full result after example-based fine-scale correction without and with textured rendering, comparison with real-world photograph of the same facial pose. Third and fourth rows: more comparison results, with asymmetric deformations not present in the input examples.*

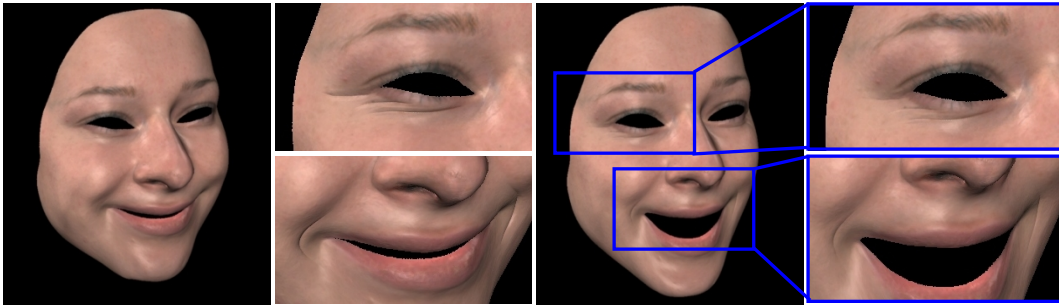


Figure 5.8: *Wrinkle Editing on an Actress' Performance.* Given an example pose without wrinkles (left), we created wrinkles under the eyes and on the cheeks (center-left). The added wrinkles blend seamlessly during the rest of the performance, as shown in the two rightmost images and the accompanying video.

Wrinkle Editing. Our face animation method also allows for very simple and intuitive editing of fine-scale details, thanks to the compact pose-space representation of details using very few example poses. Figure 5.8 shows wrinkle editing of an actress' performance. We used a basis with $P = 4$ poses, and modeled artificial wrinkles on 2 of them. With our hybrid face animation method, these added wrinkles are interpolated in pose-space, and seamlessly blend during the complete performance, in real-time.

Fine-Scale Transfer. Face models in games, virtual reality, or computer vision often look unrealistically smooth because high-resolution acquisition or manual modeling can be tedious and expensive [Golovinskiy et al., 2006]. Our method allows simple transfer of facial details on novel, animated faces, enhancing their visual plausibility. In Figure 5.9 and the accompanying video, we show that it can be applied to a wide range of input data by transferring the acquired wrinkles of the subject shown in Figure 5.7 to a female face performance captured using only sparse motion capture data, a manually designed human face rigged in Maya, and a publicly available cartoonish character rigged in Blender.

5.6 Discussion

In this chapter, we have presented a face animation method able to produce complex wrinkling effects on high-resolution meshes in real-time. Its power



Figure 5.9: *Fine-Scale Transfer.* Given the example poses shown in the first row of Figure 5.7, its details are transferred to a manually modeled human character (left), and a publicly available cartoonish character rigged in Blender (right).

stems from a two-scale representation of facial deformations, and tailored approaches for computing the deformation at each scale. A constrained deformation approach allows for overall good approximation of facial expressions by smoothly interpolating a sparse set of handles. On the other hand, an example-based deformation approach allows for real-time correction of expressive details. In fact, this model builds on our representation for facial geometry and motion presented in Chapter 3. While the deformation model of the large-scale motion is mathematically exactly the same, the computation of smaller non-linear facial features is significantly accelerated by either pre-computing or capturing example deformations and then using a learning-based approach for real-time interpolation.

Along with its hybrid nature, another strength of our solution to face animation lies on the strain-based feature vector for pose representation. [Wu et al., 1996] already related wrinkle formation to underlying lateral skin strain, and we learn the relationship in our PSD model, but it would be interesting to further study its non-linearity and the range of support. Our definition of feature vector has even proved competitive for very fast blend shape control.

Another advantage of our model is the simple fine-scale transfer to novel face models and animations that lack facial details. As future work, it would be interesting to build up a database of skin details across different gender, age, and skin types, that allows fast and intuitive visual enhancements of face animations. Furthermore, our method should theoretically be directly applicable to even higher-resolution data that contains very small wrinkles and pores.

Our face animation method trades computational complexity by memory re-

quirements, as each face vertex needs to store handle responses and learned pose displacements. Similar to recent work [Meyer and Anderson, 2007], it would be interesting to investigate more compact representations, including methods for finding optimal sets of examples and handles.

However, the method also presents some limitations. Our model could suffer from self-intersections, especially if the user edits the handles to arbitrary locations, but we did not encounter such problems when the motions are restricted to facial expressions. Anatomical accuracy would also require the addition of other important facial features like hair, teeth, or the eyes, which are outside the scope of skin deformation, but play a key role in the realism of the full face animation.

The most important limitation, common to all face animation methods driven by handles (e.g., mocap markers) is that the face model cannot react to forces, only to position constraints. Anatomical models can be a solution to this problem, as they react to external forces and can produce realistic deformations beyond facial expressions; however, to date they do not provide detailed deformations in real-time. A related benefit of anatomical models is the possibility to handle collisions naturally. To address these issues, we present in Chapter 7 a novel data-driven representation and modeling technique for simulating non-linear heterogeneous soft tissue. This technique employs finite element methods and exploits a set of measured example deformations of real-world objects, similar to the technique presented in this chapter.

Measuring Deformation Behavior

Related to facial geometry and motion, an important aspect of human faces is the behavior of soft tissue. Understanding and being able to model its complex behavior is a key ingredient for physically-based simulation of human faces. To study and acquire the physical deformation behavior of objects, we designed and built custom measurement devices. This chapter motivates design choices, presents a vision system for tracking surface deformations, and introduces our force measurement devices. We present two systems, a small portable device designed for manual use, and a fully automatic system driven by a robot arm.

These systems are used as acquisition devices for our data-driven representation and modeling technique for simulating non-linear heterogeneous soft tissue presented in Chapter 7 as well as for measurement and validation purposes when designing and fabricating materials as described in Chapter 8.

6.1 Requirements

Acquiring the mechanical properties of soft tissue is a challenging task because tissue properties are complex and usually heterogeneous, anisotropic,

6 *Measuring Deformation Behavior*

and show a non-linear relationship between applied force and resulting displacements. To obtain proper parameters of individual tissue types, usually *in vivo* or *ex vivo* measurements have been performed intra-operatively and under sterile conditions on living organs [Nava, 2007]. This is not an option to obtain an actor-specific face model. To provide a model that is capable of reacting to external forces and handle collisions naturally we are therefore restricted to non-invasive surface stimuli.

To acquire the elastic mechanical properties of an object, we propose a simple measurement device to acquire the displacement of the object relative to its rest position in combination with applied boundary conditions. Chapter 7 shows how this data is processed, and how model parameters and a compact representation as a non-linear stress-strain relationship in a finite-element model are obtained.

Design choices were

- ▶ **Simplicity and Reproducibility.** The measurement system should be simple and reproducible. We use commodity of-the-shelf cameras and force sensor hardware, and available libraries for camera and force sensor calibration.
- ▶ **In-vivo measurements and safety.** Measurements should have as little impact as possible on the actor. Techniques such as computer tomography or radiography could have negative health impacts and are not applicable for entertainment purposes. We therefore decided to use a passive vision system and manually operated force probe for acquiring properties of the facial tissue.
- ▶ **Robustness and Correspondence.** Applying force loads at arbitrary locations implies that probes are in contact with the face or object, potentially interfering with lighting and blocking camera views. The vision system has to be robust regarding interference factors. Furthermore, the system has to provide surface displacements in correspondence to the neutral, undeformed surface as deformation states in solid elastostatic mechanics are usually represented in the Lagrangian description, relative to the neutral pose.

6.2 **Hand-held Device**

We developed a simple data acquisition system consisting of force probes and a marker-based trinocular stereo system. Deformations are induced by

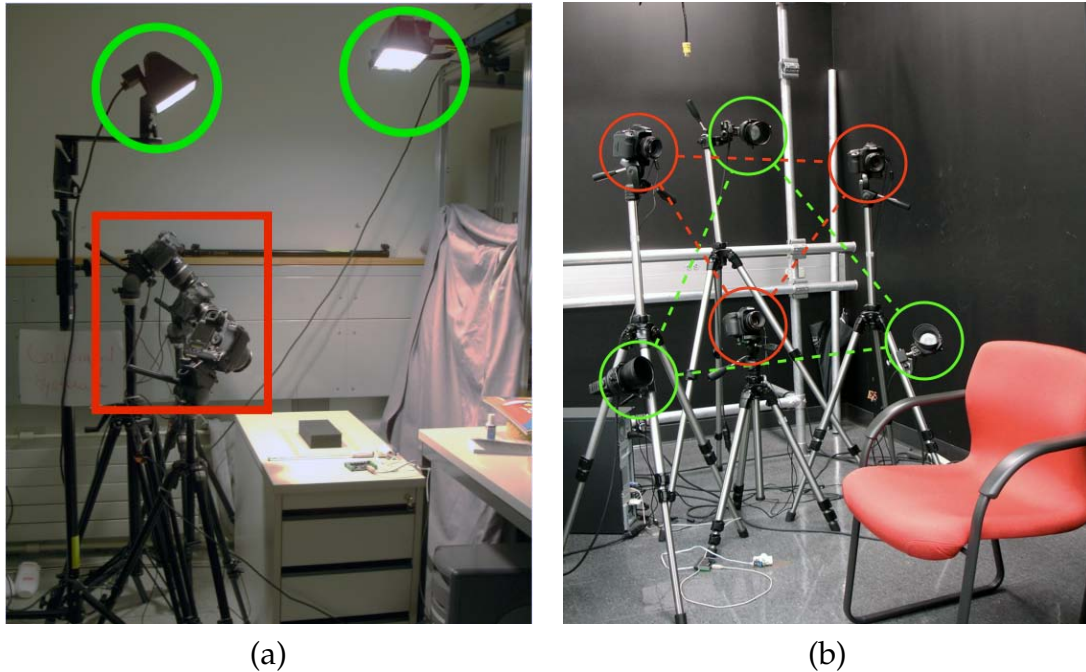


Figure 6.1: *Trinocular Stereo Vision System.* Our trinocular stereo vision system consists of three high-resolution cameras (indicated in red) and two to three light sources (indicated in green). The cameras are arranged in a triangular setup, which helps maximize visibility during capture of a contact interaction. The light sources ensure uniform illumination during the acquisition.

physical interaction with the object. We decided to use a marker-based system due to its simplicity, robustness, and independence of the object’s surface properties.

6.2.1 Trinocular Stereo Vision System

Figure 6.1 shows our trinocular stereo vision system, consisting of three Canon 40D cameras that capture images at a resolution of 3888×2592 . These cameras are placed in a triangular configuration to minimize occlusions caused by the contact probes during data acquisition. We built an external trigger device to synchronize the three cameras, and use additional light sources to ensure uniform illumination during the acquisition process. The surface displacement during static deformations is measured using a set of markers that we paint on the object’s visible surface. Our system is ca-

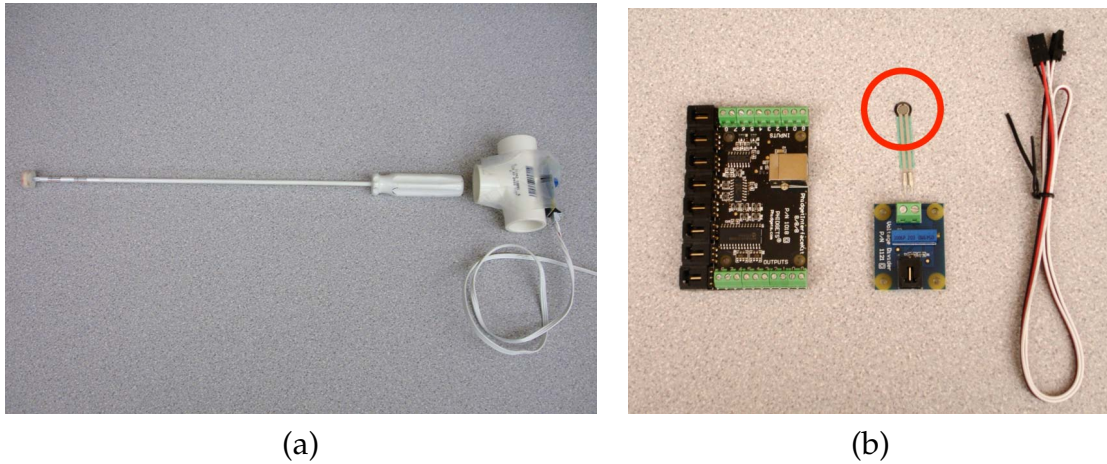


Figure 6.2: (a) Contact probe with integrated force sensor. (b) From left to right: USB Interface Kit, Force Sensing Resistor (red circle), Phidget Voltage Divider, and connection cable.

pable of measuring viewpoint-registered marker positions to an accuracy of < 1 mm.

We built contact probes with arbitrary shapes and circular disks of different diameters attached to the tip of a long screwdriver (see Figure 6.2). We estimate the position and orientation of the contact probe using two markers on the white shaft of the screwdriver. To measure the magnitude of the contact forces we use a 0.2 inch Force Sensing Resistor (FSR) (Item S-20-1000-FS2) connected to a Phidget Voltage Divider (Item S-50-P1121) and USB Interface Kit 8/8/8 (Item C-200-P1018) by Trossen Robotics. The force sensor's read operation is synchronized with the external camera trigger signal.

6.2.2 Processing Pipeline

The processing pipeline consists of three basic steps for reconstruction the surface deformation. First, the marker positions are identified in the images. Then, we perform a 3D reconstruction of the marker positions. Finally, the marker positions are registered with a template mesh to provide correspondence and to interpolate lost markers that could not have been reconstructed due to occlusions.

We use simple thresholding in the CIELab color space followed by a correction step using morphological operations for all the different marker colors

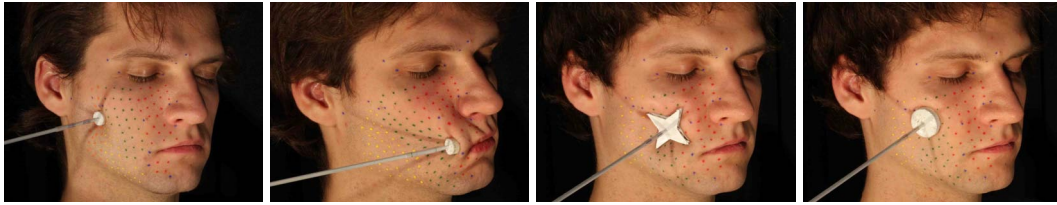


Figure 6.3: *Measured facial deformations. In total, we acquired more than 70 different interaction scenarios with varying contact shapes and locations, each under 4 increasing force loads. This figure shows a subset of them.*

independently. This simple procedure provides a robust identification of the marker positions in the image domain.

Marker positions in 3D are reconstructed applying a simple stereo matching. A marker can be reconstructed if it is visible in at least two of the three camera images. These reconstructed markers are then registered to a template mesh by subtracting the general rotation and translation component of the head movement. We estimate this rotation using a modified version of Horn’s shape matching algorithm [Horn, 1987], giving markers far away from the deformation location a high weight. In rare cases markers cannot be reconstructed due to occlusions. In this case, we employ a linear shell-based approach [Botsch and Sorkine, 2007] which incorporates the prescribed displacements (visible markers) as boundary constraints of the template mesh, and, otherwise, minimizes surface stretching and bending to estimate the displacement of the occluded markers. This yields a complete reconstruction of all markers.

In total, using this system, we acquired more than 70 different interaction scenarios with varying contact shapes and locations, each under 4 increasing force loads. Figure 6.3 shows of subset of measured facial deformations.

6.3 Automated Setup

To acquire surface deformations of objects with a wide range of material properties we built an automatic measurement system that is able to acquire many different materials with varying geometry and surface properties. We use our system to probe base material samples, combinations of base materials for model validation, complex objects that we would like to reproduce, and their printed counterparts for validation purposes as described in detail

6 Measuring Deformation Behavior

in Chapter 8. The main advantage of the automated setup is the ability to accurately reproduce the applied forces and sample the object exactly according to a planned sampling strategy, whereas the previously presented hand-held device is more versatile and can be safely applied to humans.

Our fully automated measurement setup (Figure 6.4) consists of a four DOF robot arm (from MicroProto Systems), a six-axis force-torque sensor (Nano 25 from ATI), and a vision subsystem to track surface displacements. All the subsystems are integrated and controlled from a single PC. The resolution of the robot arm is 0.003 mm and its repeatability is 0.01 mm. The maximum range of the force sensor is 125 N with a resolution of 1/24 N. This system allows to repeatably probe materials placed and fixed on a rigid stage and to accurately determine the corresponding exerted forces.

6.3.1 Processing Pipeline

The vision subsystem consists of seven high-resolution Basler Pilot cameras running at a resolution of 1600 × 1200 pixels. We set up the cameras on a half-circle above the robot arm to minimize occlusions and added diffuse lighting. The cameras are calibrated using the toolbox developed by [Svoboda et al., 2005].

We paint marker dots on the surfaces of our objects in a regular grid with 3 mm spacing. The marker positions are extracted using a scale and affine invariant blob detector ([Mikolajczyk and Schmid, 2004]) and tracked in all the frames. We then reconstruct the 3D marker positions using triangulation. In contrast to the hand-held device described in Section 6.2, we continuously acquire images at a sufficiently high frame rate to reliably track markers between consecutive frames. No template fitting is required. For each acquisition we use 30 to 200 deformation steps depending on the stiffness of the material. The maximal forces are in the range of 35 to 50 N. Figure 8.4 shows a few measured materials. The tracked markers and corresponding forces are finally registered to an initial, undeformed surface mesh.

Examples of performed measurements using this system are shown in Section 8.5 and Section 8.6.

6.4 Discussion

In this chapter, we have presented two systems for acquiring force - surface displacement pairs of deformable objects. For computing surface displace-

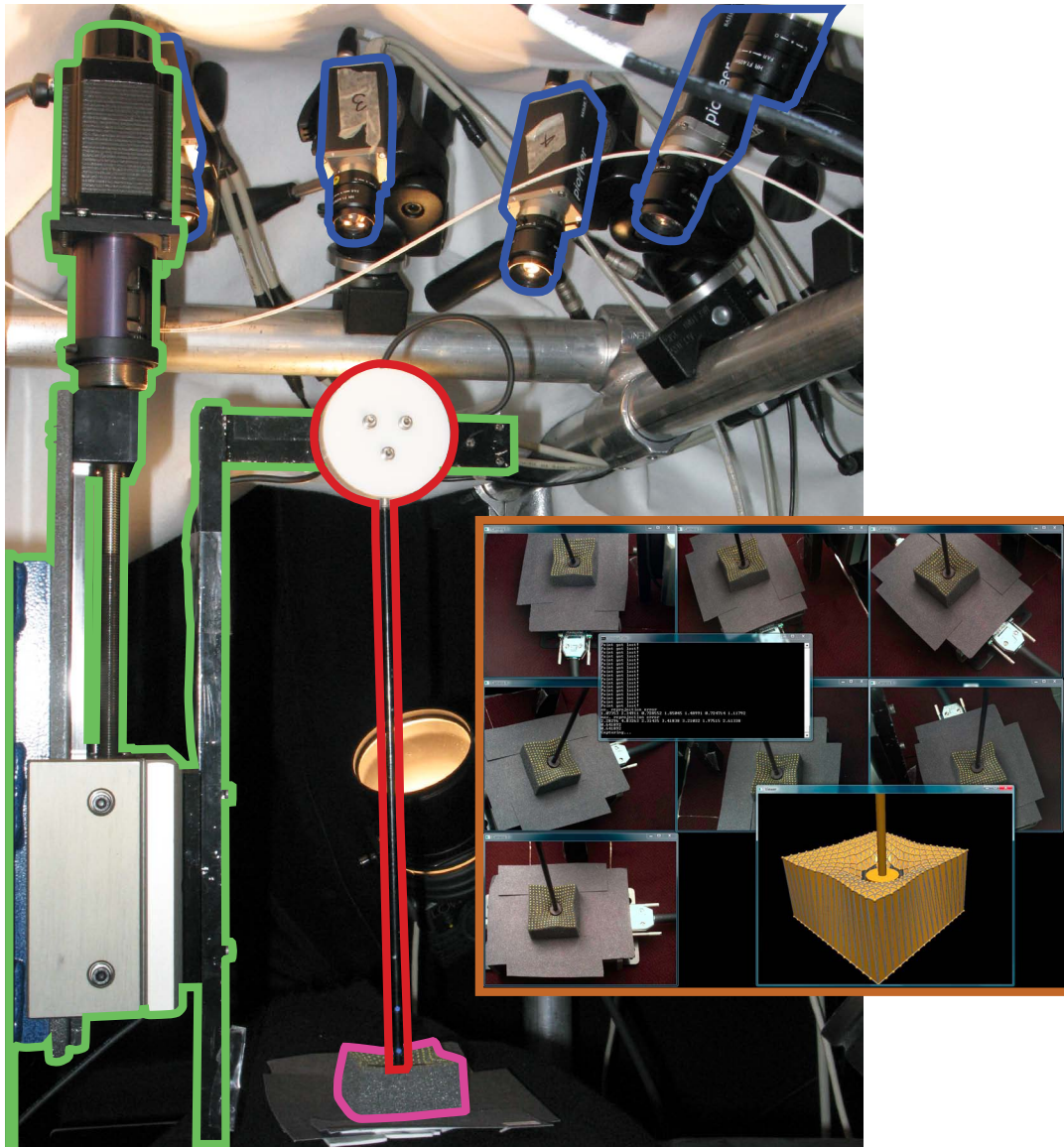


Figure 6.4: *The automated system for measurement of material deformations consists of cameras (blue), a robot arm (green), and a force sensor attached to a stick (red). A sample material block is shown in pink and the inset shows a screen shot of our processing software.*

6 *Measuring Deformation Behavior*

ments and establish correspondence over time, we applied a dot pattern texture on the object's surface and employed standard computer vision techniques for 3D reconstruction and tracking. The deformation is induced by a round probe with an attached force sensor which can be either used as a hand-held device or in combination with a robot arm.

Standard state-of-the-art constitutive material parameter estimation devices such as for example tensile, compression, or biaxial testing devices come with special requirements regarding the design of clamps and shape of the specimen because forces must be introduced uniformly whilst keeping the clamping interface intact [Hannon and Tiernan, 2008]. These requirements are usually difficult to fulfill and impractical. In contrast, our approach does not assume a regular force or strain field at the interface. However, this comes at the cost that the inverse problem of estimating material parameters is much more challenging and requires running iteratively a full non-linear FEM simulation including optimization strategies. Details on the material model, FEM simulation, and fitting are described in Sections 7.2, 7.3, 8.3, and in Appendix B.

In addition to measuring the applied force and resulting surface displacement, acquiring and modeling the remaining boundary conditions is crucial. For the acquired material samples measured in Chapter 7 and Chapter 8, we usually place the specimen on a rigid bench and specify Dirichlet boundary conditions at the contact areas of bench and specimen. To ensure zero displacements in these regions, we observed that it is necessary to either glue or 3D print these areas on a thin, rigid surface. Furthermore, modeling the force distribution at the contact area of the force tip is complex. A study showing the non-uniform stress distribution underneath a axisymmetric indenter based on finite element calculations can be found in [Hollenstein, 2008]. As the resolution of our force sensor is limited, we model the indenter as an analytic surface with high friction and estimate in-plane forces during the finite element simulation. As shown based on side-by-side comparisons and error evaluations in Section 8.6, this approach resembles the force probe - specimen interface well. Currently, due to the cost as well as the physical size of commercially available force sensors, manufacturing of spatially high-resolution force sensor arrays is impractical, but could be an interesting research area and future tool for more accurately acquiring force distributions in contact areas.

Modeling of Non-Linear Heterogeneous Soft Tissue

In this chapter, we present a novel data-driven representation and modeling technique for simulating non-linear heterogeneous soft tissue that simplifies the construction of convincing deformable models. We employ finite element methods and exploit a set of measured example deformations of real-world objects acquired by the measurement system presented in Chapter 6. Every measured example deformation is transferred into a local element-wise strain space, and represented as a locally linear sample of the material's stress-strain relation (Section 7.2). The full non-linear behavior is modeled by interpolating the material samples in strain space using radial basis functions (RBFs). Finally, we will show in shown in Section 7.4, that a simple elastostatic finite-element simulation of the non-linearly interpolated material samples based on incremental loading allows for efficient computation of rich non-linear soft-tissue simulations. This material model also serves as one of the main building blocks for designing and fabricating materials with desired deformation behavior in Chapter 8.

7.1 Introduction

Numerous researchers have combined Newtonian mechanics, continuum mechanics, numerical computation and computer graphics, providing a powerful toolkit for physically-based deformations and stunning simulations, with application in feature films, video games, and virtual surgery, among others. However, achieving realistic soft-tissue deformations requires careful choices for material models and their parameters. Many real-world objects consist of *heterogeneous materials*, requiring spatially varying material parameters such as Young's modulus and Poisson's ratio. Setting them is a difficult and time-consuming process. Even more challenging is the problem of *material non-linearities*. Most materials, for example rubber or biological soft tissue, show non-linear constitutive behavior, i.e., a non-linear relationship between stress and strain. Despite the wide variety of non-linear constitutive models in the literature, such as the popular hyperelastic Neo-Hookean and Mooney-Rivlin models [Ogden, 1997], this is still an active research area in material science. Nonetheless, non-linear physics equations are often simplified approximations to real material behavior, and choosing the appropriate model as well as tuning its parameters are extremely complex tasks.

We present a novel data-driven representation and modeling technique for simulating non-linear heterogeneous soft tissue that simplifies the construction of convincing deformable models (Fig. 7.1). Our technique employs

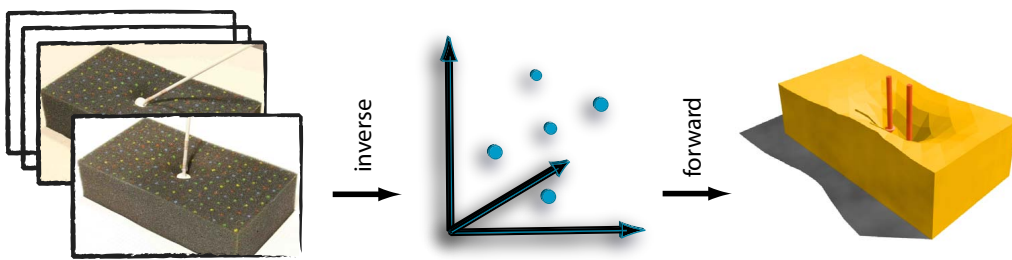


Figure 7.1: *Acquiring and modeling non-linear quasi-static soft tissue behavior. From left to right: An object is probed with a force sensor to acquire several example deformations, the applied force direction, and the force magnitude. For every measurement we estimate its stress-strain relationship and represent it as a sample in strain space. During runtime, we interpolate these samples in strain space using radial basis functions (RBFs) to synthesize deformations for novel force inputs.*

finite element methods and exploits a set of measured example deformations of real-world objects, thereby avoiding complex selection of material parameters. We transfer every measured example deformation into a local element-wise strain space, and represent this example deformation as a locally linear sample of the material's stress-strain relation. We then model the full non-linear behavior by interpolating the material samples in strain space using radial basis functions (RBFs). Finally, a simple elastostatic finite-element simulation of the non-linearly interpolated material samples based on incremental loading allows for efficient computation of rich non-linear soft-tissue simulations.

Other earlier work in computer graphics and robotics also proposed measurement-based model fitting as a means for obtaining deformable object representations [Pai et al., 2001, Lang et al., 2002, Schoner et al., 2004], but was limited to linear material models with global support. In contrast, our work is the first to represent *complex non-linear heterogeneous materials* through spatially varying non-linear interpolation of local material properties. Our complete soft tissue capture and modeling pipeline is also distinct for its simplicity. Based on a simple-to-build capture system consisting of force probes and marker-based trinocular stereo (Chapter 6), we present an efficient and robust algorithm for fitting the local strain-space material samples. We demonstrate the effectiveness of our soft-tissue capture and modeling method for several non-linear materials and biological soft tissue. The combination of simplicity and efficiency, both in acquisition and computation, and the high-expressiveness of the results make our technique applicable for interactive applications in computer graphics and other fields.

7.2 Modeling of Non-Linear Materials

In this section, we describe our representation of non-linear heterogeneous elastic materials, and how this representation is used for modeling soft tissue deformations. We first give an overview of the representation, and then describe how we parameterize the materials and how this parameterization extends from the continuum setting to a finite element discretization. We also explain how we support material non-linearities through interpolation of local linear models, and finally we describe our algorithm for computing non-linear elastostatic deformations based on incremental loading.

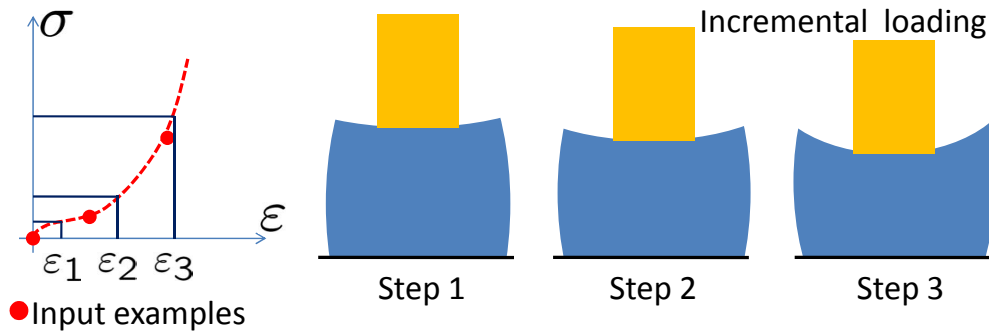


Figure 7.2: As a pre-process, we identify material parameters for a set of input deformations as samples of the element-wise strain-stress curve (red dots). At run-time, we model non-linear deformations through incremental loading. Given the strain for the current loading step, we obtain the material parameters through interpolation of the input samples, and solve an FEM elastostatic simulation.

7.2.1 Overview of our Approach

In materials science, (one-dimensional) elasticity properties have long been described through stress-strain curves. Inspired by this popular representation, we opt for modeling three-dimensional elastic properties by sampling the stress-strain function at various operating regimes and interpolating these samples in strain-space (See Figure 7.1).

More specifically, we characterize each sample of the stress-strain function using a (local) linear constitutive model. Then, in order to capture material non-linearity, we define the parameter values of the constitutive model at an arbitrary operating point through scattered-data interpolation in strain-space. Moreover, in order to capture material heterogeneity, we compute both the stress-strain samples and the scattered-data interpolation in a spatially varying manner. Figure 7.3 shows example deformations with color-coded Young's modulus, which varies both as a function of the location and the local strain.

It is worth noting that our model can capture elasticity properties, but not plasticity or viscosity, among others. Our model builds on FEM and linear elasticity theory, and we refer the interested reader to books on the topic [Bathe, 1995, Hughes, 2000].

7.2.2 Discretization and Parameterization

We use linear co-rotational FEM to locally represent a deformable object's elastic properties. In other words, given an object's deformed configuration, we model the stress-strain relationship with linear FEM. We capture non-linearity by varying the parameters of the stress-strain relationship as a function of the strain itself. Given a displacement field \mathbf{u} , the linear co-rotational FEM employs Cauchy's linear strain tensor

$$\varepsilon(\mathbf{u}) = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right). \quad (7.1)$$

Invariance of the strain under rotations is obtained by extracting the rotational part of the deformation gradient through polar decomposition, and then warping the stiffness matrix [Müller and Gross, 2004].

Thanks to symmetry of the strain and stress tensors, we can represent both as 6-vectors. Given the strain tensor, we construct the 6-vector as

$$\varepsilon = (\varepsilon_{xx} \ \varepsilon_{yy} \ \varepsilon_{zz} \ \varepsilon_{xy} \ \varepsilon_{xz} \ \varepsilon_{yz})^T, \quad (7.2)$$

and similarly for the stress. The local linear material yields then a relationship

$$\sigma(\mathbf{u}) = E\varepsilon(\mathbf{u}) \quad (7.3)$$

between strain and stress. For each element (in our case, a tetrahedron), assuming locally linear isotropic material, the 6×6 stress-strain relationship matrix E can be represented by Young's modulus E and Poisson ratio ν

$$E = \frac{E}{(1+\nu)(1-2\nu)} (G + \nu H), \quad (7.4)$$

with the two constant matrices

$$G = \text{diag}(1, 1, 1, 0.5, 0.5, 0.5) \quad (7.5)$$

and

$$H = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad (7.6)$$

This parametrization is intuitive, where the Poisson ratio ν is unit-less and describes material compressibility, while Young's modulus E defines material

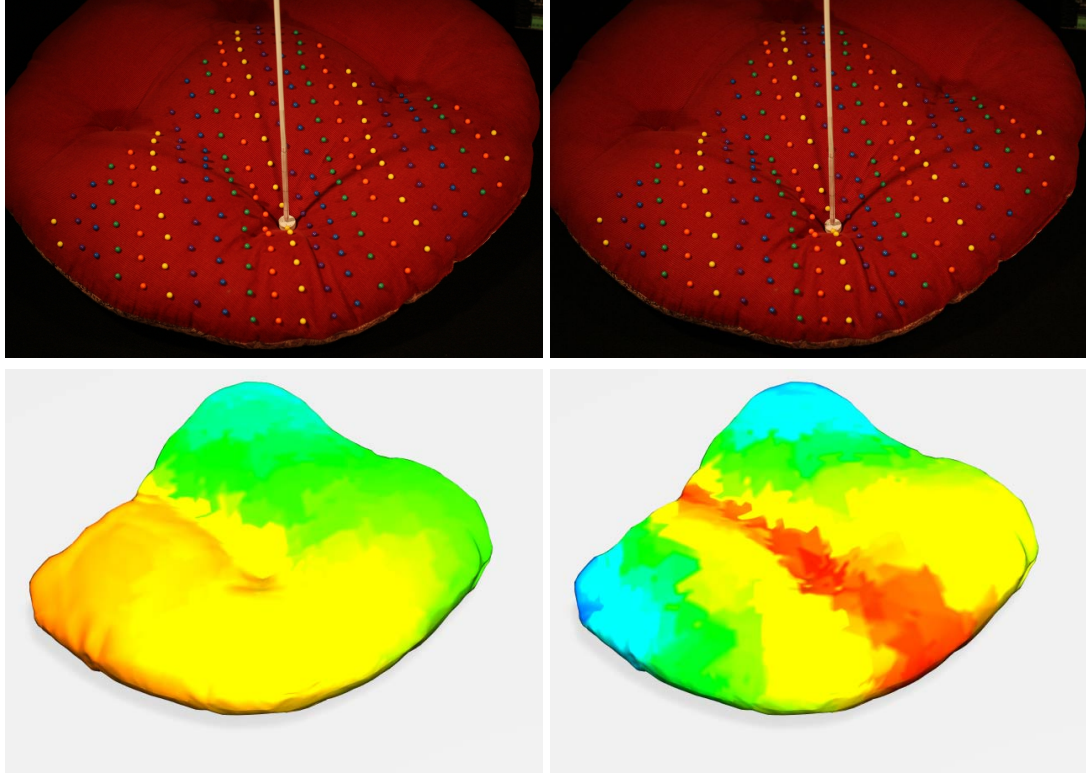


Figure 7.3: *Two examples of a deformed pillow with color-coded Young's modulus ('blue' is low, 'red' is high), which varies both as a function of location and the local strain. Probe pressure was higher on the right.*

elasticity. However, we employ an alternative parameterization (λ, α) that allows us to describe the stress-strain relationship as a linear function of the parameters [Becker and Teschner, 2007]:

$$E = \lambda \mathbf{G} + \alpha \mathbf{H}, \quad (7.7)$$

with

$$\lambda = \frac{E}{(1 + \nu)(1 - 2\nu)} \quad \text{and} \quad \alpha = \lambda \nu. \quad (7.8)$$

The parameter α is also known as Lamé's first parameter in elasticity theory, whereas λ is not directly related to any elasticity constant. With the (λ, α) parameterization, the stiffness matrix and the elastic forces become linear in the parameters. We exploit this property in our parameter fitting algorithm in Section 7.3.1.

The per-element stiffness matrix can be written as

$$K_e = \lambda_e V_e B_e^T G B_e + \alpha_e V_e B_e^T H B_e, \quad (7.9)$$

where V_e is the volume of the element (i.e., tetrahedron), and B_e is a matrix dependent on the initial position of the element's nodes. The complete stiffness matrix is obtained by assembling the warped per-element stiffness matrices $R_e K_e R_e^T$, where R_e is the element's rotation. By grouping all material parameters $\{\lambda_e, \alpha_e\}$ in one vector \mathbf{p} , the stiffness matrix is parameterized as $K(\mathbf{p})$.

7.2.3 Strain-Space Interpolation

As introduced earlier, we describe the non-linear material properties through scattered-data interpolation of known local linear parameters in an element-wise manner. We obtain these known local parameters from a set of example deformations, largely simplifying an artist's job of tuning material parameters for complex non-linear constitutive models.

Let us assume a set of M known example measurements, each with a corresponding element-wise strain vector $\varepsilon_i \in \mathbb{R}^6$ and a parameter vector $\mathbf{p}_i = (\lambda_i, \alpha_i)^T$. Recall that we use a rotationally-invariant strain by extracting the rotation of the deformation gradient through polar decomposition [Müller and Gross, 2004]. Our non-linear strain-dependent material $\mathbf{p}(\varepsilon)$ is formed by interpolating linear material samples $\mathbf{p}_i(\varepsilon_i)$. At a given deformed configuration, the non-linear material is represented by the corresponding linear material that achieves the same force-displacement relationship. Note that we do not exploit linearization in the more traditional way of capturing the local slope of a non-linear function.

For each element, we define the stress-strain relationship through scattered-data interpolation in the strain-space \mathbb{R}^6 using radial basis functions (RBFs). The element-wise function describing the material, $\mathbf{p}(\varepsilon) : \mathbb{R}^6 \rightarrow \mathbb{R}^2$, has the form

$$\mathbf{p}(\varepsilon) = \sum_{i=1}^M \mathbf{w}_i \cdot \varphi(\|\varepsilon - \varepsilon_i\|), \quad (7.10)$$

where $\varphi(\cdot)$ is a scalar basis function, and $\mathbf{w}_i \in \mathbb{R}^2$ and ε_i are the weight and feature vector for the i 'th measurement, respectively. We employ the biharmonic RBF kernel $\varphi(r) = r$. This globally supported kernel allows for smoother interpolation of sparsely scattered example poses than locally supported kernels, and avoids difficult tuning of the support radius [Carr et al., 2001].

As a preprocess, we compute the RBF weights \mathbf{w}_i . This reduces to solving $2T$ linear $M \times M$ systems for a deformable object with T elements due to the fact that the stress-strain relationship is an element-wise description of the

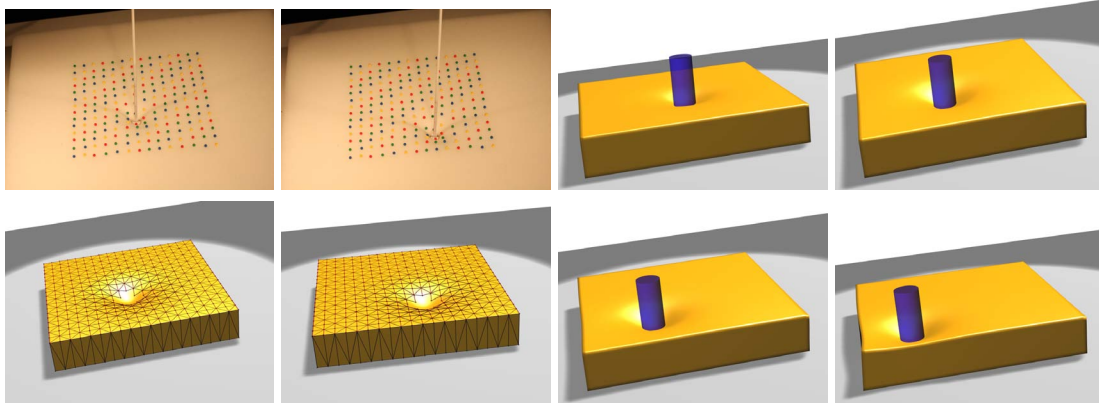


Figure 7.4: *Two left-most columns: Comparison of captured and synthesized deformations for a foam block. Two right-most columns: Examples of interactive deformations produced by sliding a cylinder on top of the model.*

material. This also leads to scattered-data interpolation of the material parameters in a rather low-dimensional \mathbb{R}^6 domain. In contrast, interpolation of material properties is much more complicated in earlier approaches based on linear models with global support [Pai et al., 2001] due to the extremely high dimensionality of the parameterization.

7.2.4 Elastostatic FEM Simulation

We compute novel deformations using an elastostatic FEM formulation $K\mathbf{u} = \mathbf{F}$, where the force \mathbf{F} includes, among others, the load produced by a contact probe. To correctly capture the material's non-linearity during the deformation, we apply the load of the probe gradually, and solve the elastostatic FE problem for each load increment. In other words, at each loading step we measure the current strain ε , we compute the material parameters $\mathbf{p}(\varepsilon)$ by means of the interpolation described above, we formulate the elastostatic problem, and we solve it for the new deformations. The incremental loading procedure ensures that the non-linearity of the material is correctly captured during the complete deformation process, with the material parameters depending on the strain at all times.

For contact handling, we compute a distance field for the rigid probe object that produces the deformations. We test for collisions between points on the deformable object and the distance field and, upon collision, we compute

the penetration depth and direction. We then define a linear force field at each colliding point and solve the FEM simulation through iterative quasi-static simulation. At each iteration of the quasi-static FEM simulation, we first compute the material parameters for the current configuration based on the interpolation algorithm described above. Then, given the stiffness matrix and the linear collision force field, we define a quasi-static problem and solve for the new displacements. We compute several iterations until an equilibrium is reached.

7.3 Fitting the Material Parameters

We now describe how we compute the actual material parameters for a given object. This consists of two parts: First, estimating parameter values for each deformation example, and second, selecting a suitable basis from all the deformation examples.

7.3.1 Parameter Estimation Algorithm

In order to estimate a sample of the stress-strain relationship, we apply a known input force to the object. For each captured deformation we can distinguish three different regions on the object's surface:

- ▶ the probing region, with measured non-zero forces and measured displacements,
- ▶ the attached region, with unknown forces and zero displacements, and
- ▶ the free region, with zero forces and measured displacements.

We use $\bar{\mathbf{x}}$ and $\bar{\mathbf{F}}$ to denote the vectors of known displacements and forces, respectively, at the points corresponding to mesh nodes in the model.

Given measured displacements and forces, we compute spatially varying material parameters \mathbf{p} as:

$$\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} \left\{ \sum_{i=1}^n \|\mathbf{x}_i(\mathbf{p}, \bar{\mathbf{F}}) - \bar{\mathbf{x}}_i\|^2 + \gamma \|L\mathbf{p}\|^2 \right\}, \quad (7.11)$$

where $\mathbf{x}_i(\mathbf{p}, \bar{\mathbf{F}})$ denotes the position of a mesh node as a function of material parameters and the measured forces. The sparse Laplacian matrix L enforces spatial smoothness of parameters. We employ the umbrella operator

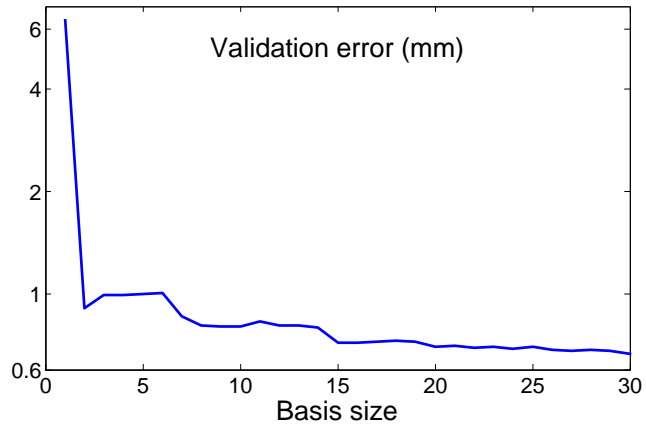
[Zhang, 2004] $(L\mathbf{p})_i = \sum_j w_{i,j}(\mathbf{p}_i - \mathbf{p}_j)$, where i and j refer to tetrahedron labels, and $w_{i,j} = 1$ iff two tetrahedra share a vertex. This regularization is required to prevent overfitting due to noise in the acquired data. This is also mathematically required to obtain a well-posed problem because the number of parameters is always twice the number of tetrahedra, $|\mathbf{p}| = 2T$, whereas the number of measured positions $|\bar{\mathbf{x}}| = n$ may be smaller, which would result in an underconstrained problem. We also considered scattered data interpolation of material parameters in object space as an alternative for addressing the underconstrained problem, but it would be difficult to decide where to place the samples for highly heterogeneous objects.

We minimize the non-linear residual Equation (7.11) iteratively using the Levenberg-Marquardt algorithm [Levenberg, 1944]. We derive the Jacobian matrix in the Appendix. Instead of defining the residual in terms of measured positions, the error functional could also be described in terms of measured forces [Becker and Teschner, 2007], yielding a linear optimization problem. However, our observations have shown that this approach is unstable when the force-displacement relationship is not close to linear material behavior.

7.3.2 Strain-Space Basis Selection

A material capture session consists of capturing N example deformations, from which we obtain the training dataset of N parameter vectors for each element in the mesh. However, this dataset may be rather large, and we are interested in selecting a compact set of M basis parameter vectors for each element. Note that M need not be the same for all elements.

We select the basis in the same greedy manner as proposed by [Carr et al., 2001]. We start by setting a parameter vector at zero strain with the average parameters computed for very small-strain deformations. We then add the parameter vector with largest error, until a given error tolerance is achieved. After each parameter vector is added to the basis, we need



to compute the RBF weights that best fit the parameter vectors for all N example deformations in a least-squares manner, as described in Equation (7.10).

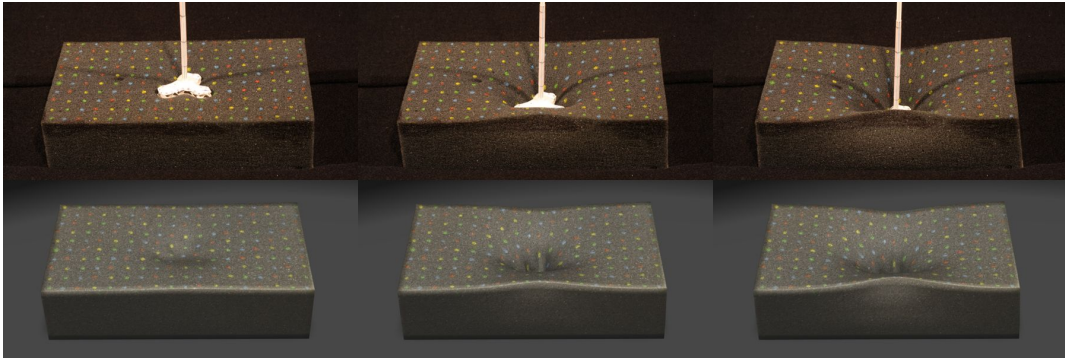


Figure 7.5: Comparing real (top) and modeled (bottom) deformations with a different contact probe than the one used in the data acquisition phase.

This figure shows the evolution of the fitting error for the foam block in Figure 7.4. This error plot accumulates the error for all captured deformations, not only those added to the basis. The error drops quickly after adding the second parameter vector to the basis because the first vector may not represent the average material behavior well. See Section 7.4 for more details on the validation of our method.

7.4 Results

We have evaluated the quality of our soft-tissue capture and modeling technique on several real-world objects, including two foam blocks, a heterogeneous soft pillow, and a human face. Furthermore, we analyzed and compared the path-dependence of our method to existing element formulations.

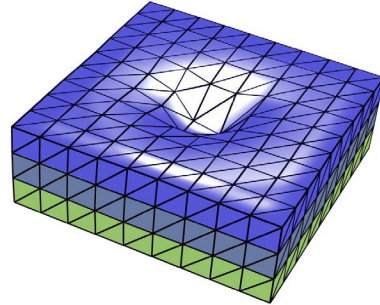
7.4.1 Model Evaluation

Figure 7.4 shows a foam block with homogeneous material. We acquired 48 deformation examples, well distributed over the foam to induce deformations in all 1,805 tetrahedra of our model. We then constructed the non-linear material representation, with bases of 8 samples per tetrahedron on average, using the procedure in Section 7.3.1. Even though the object is homogeneous, it should be noted that the material parameters that were estimated for each input example are non-homogeneous due to non-linearities in the stress-strain relationship. The average fitting error for the captured deformations is less than 1 mm (see figure in Section 7.3.2). Figure 7.4 shows

7 Modeling of Non-Linear Heterogeneous Soft Tissue

synthesized deformations produced with our technique using a probe with a larger, different contact area than the probe used for data acquisition.

To compare our model to a uniform linear co-rotational model we use the homogeneous foam shown in Figure 7.5 and Figure 7.6. We captured 12 deformation examples with the probe near the center of the block and modeled the object with 3,240 tetrahedra. We computed an average-fit linear co-rotational model that best approximates all the input deformations. As shown in Figure 7.6, our model (blue) accurately captures the hyperelastic behavior of the foam, while the average-fit linear co-rotational model (green) underestimates the deformation at small force values and overestimates it at large ones. In addition, the linear co-rotational model suffers from element inversion for large forces.



Our model is of course not confined to the contact shapes that were used during data acquisition. Figure 7.5 shows a side-by-side comparison of our model (bottom) to real deformations (top) using a different contact probe than the circular one we used for data acquisition. We captured the applied force with the new contact probe, and then distribute it uniformly in the simulated setting. The figure shows high correspondence between the real and simulated scenarios. We refer the reader to the accompanying video for an animated side-by-side comparison.

To evaluate the sensitivity of our capture and modeling approach to measurement noise we created example deformations of a virtual block with three layers of user-defined non-linear materials.

We then evaluated the accuracy in matching these deformations with our model under different levels of noise in the input data. Specifically, we applied Gaussian noise with a variance of 10%, 20% and 30% to the input displacements and then measured the L^2 error for all deformations and error levels. On average, we obtain an error of 0.3% of the maximum displacement for the case without error, and 2.1%, 3.1% and 4.4% for the cases with 10%, 20% and 30% input noise, respectively.

Figure 7.7 shows a pillow object with heterogeneous behavior even in its rest state. The screenshots compare the captured deformations with the deformations of the 1,691 tetrahedra model synthesized with our algorithm. The figure also shows screenshots of deformations at interactive frame rates of about 10 Hz on a standard PC.

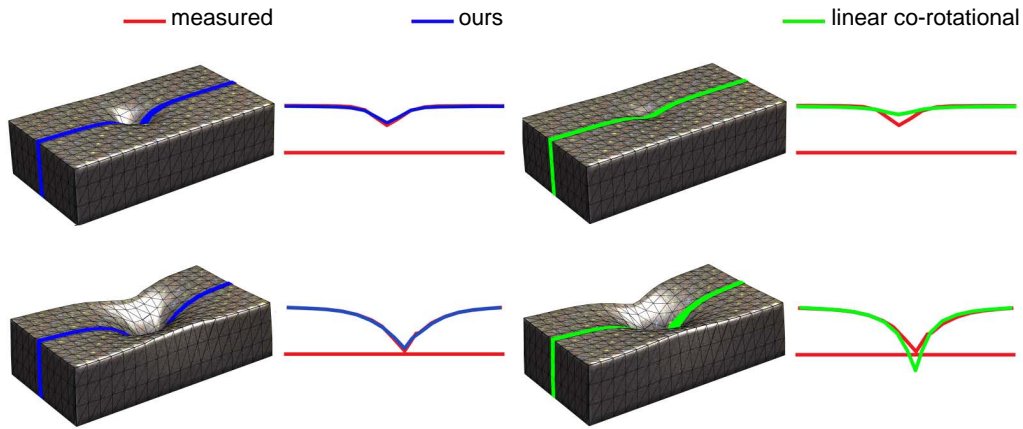


Figure 7.6: Comparison of deformations using our method vs. an average-fit linear co-rotational model.

7.4.2 Facial Deformation

We have also applied our soft-tissue capture and modeling technique to the challenging task of facial deformations, as shown in Figure 7.9. We have modeled the facial tissue with a single layer of 8,261 tetrahedra that are attached to a low-resolution skull model. To model the sliding contacts between the tissue and the skull we use the same contact handling as for the probe object (see Section 7.2.4). Given the deformation of the tetrahedral mesh, we compute the deformation of a high-resolution triangle mesh using a smooth embedding based on moving least squares interpolation like [Kaufmann et al., 2008].

Note that our face model does not correctly capture all types of deformations because we use a model with closed lips, and all the deformation examples in the training dataset were captured with relaxed muscles and closed jaw. Nevertheless, the model is able to produce compelling deformations even without anatomically correct modeling of the musculoskeletal structure of the face.

7.4.3 Test for Path-Independence

We also examined the behavior of our method and compared its properties to existing non-linear "hyperelastic" elements. In the remainder of this section,

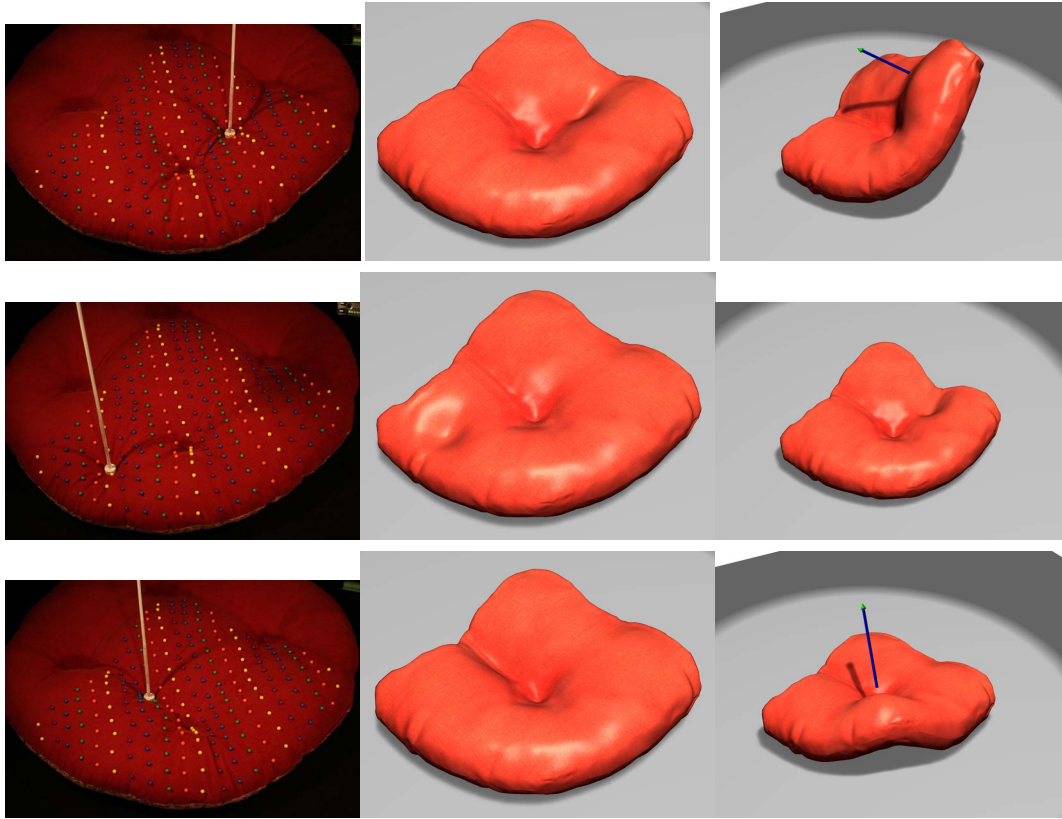


Figure 7.7: *Two left-most columns: Comparisons of captured and synthesized deformations for a heterogeneous non-linear pillow. Right column: Interactive deformations of the model produced by pushing (top) and pulling (bottom).*

we first introduce and describe a test for path-independence response of elements as suggested by [Jabareen and Rubin, 2007] and compare the reported error numbers to our results. This comprises elements available in standard computer codes such as ABAQUS, ADINA, ANSYS and FEAP as well as the Cosserat point element.

Depending on two properties, finite element formulations can be classified into hyperelastic, Cauchy elastic, and hypoelastic. These properties are:

- ▶ The nodal forces (or stresses) are functions only of the nodal positions (or deformations) of the element.
- ▶ The external work done by nodal forces (or stresses) between two states of nodal positions (or deformations) is independent of the deformation path between these two states.

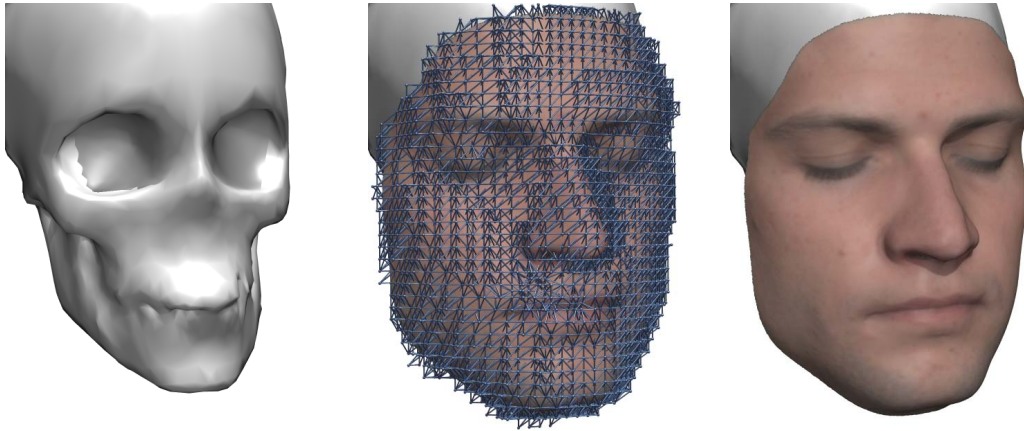


Figure 7.8: *Left: Approximate skull model; Middle: Facial tissue consisting of 8,261 tetrahedra with sliding contacts between tissue and skull; Right: Embedded high-resolution triangle mesh using moving least squares interpolation.*

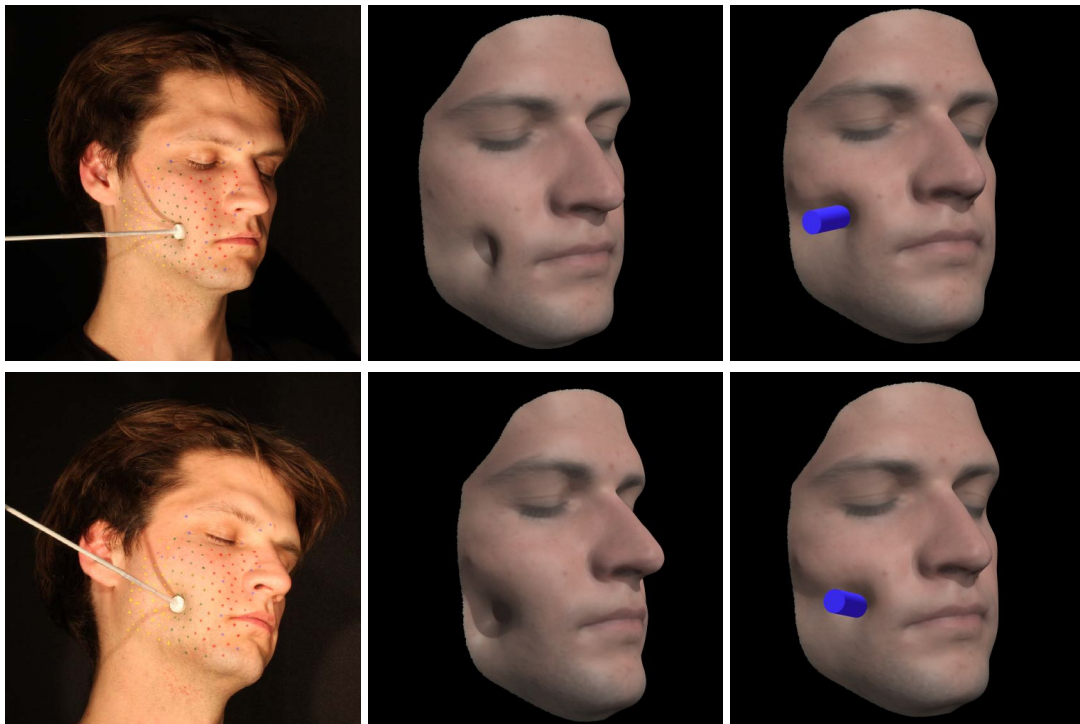


Figure 7.9: *Left: Capture of facial deformations; Middle: Synthesized deformations for the captured examples; Right: Frames of an animation with a cylindrical probe pressing on the cheek.*

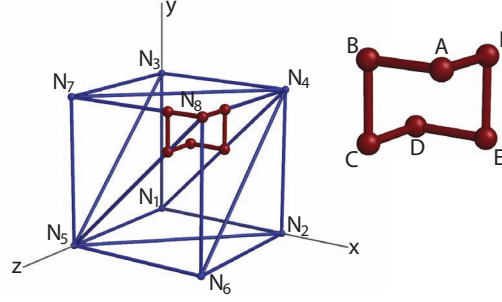


Figure 7.10: A test for path-independence. Cubical object consisting of 5 tetrahedral elements (blue) and the simulated paths (red) used to test path-independency of various element formulations [Jabareen and Rubin, 2007].

If both properties are fulfilled, the element is considered to be hyperelastic. If only the first property is fulfilled, then the element is considered to be Cauchy elastic, and if none of both properties are fulfilled, then the element is called hypoelastic [Jabareen and Rubin, 2007].

The error due to path-dependency can be specified by investigating the residual distortion after several full cycles as well as comparing relative errors on the total external work done on the object. In particular, following quantities are introduced

$$W_A^{\leftrightarrow} = W_{ABCDEFA}, W_A^{\leftarrow} = W_{AFEDEFA} \quad (7.12)$$

$$W_D^{\rightarrow} = W_{ABCD}, W_D^{\leftarrow} = W_{AFED}, \quad (7.13)$$

where $W_{I\dots J}$ is the total work done along the path $I \cdots J$. In our case, as we only prescribe the motion of one vertex, the total external work can be computed by

$$W = \int \mathbf{f}_8 \cdot d\mathbf{u}, \quad (7.14)$$

where \mathbf{f}_8 is the external nodal force at node N_8 .

In particular, we are interested in the relative error values

$$E_A^{\leftrightarrow} = \frac{W_A^{\leftrightarrow}}{W_D^{\rightarrow}}, E_A^{\leftarrow} = \frac{W_A^{\leftarrow}}{W_D^{\leftarrow}}, \text{ and } E_D = \frac{W_D^{\rightarrow} - W_D^{\leftarrow}}{W_D^{\rightarrow}}. \quad (7.15)$$

To test numerically the path-independence of our method, a node of a cubical object consisting of 5 tetrahedral elements is moved along specified deformation paths as depicted in Figure 7.10. The cube in its reference configuration is of size $(1m \times 1m \times 1m)$, the nodes $N_i, i \in \{1, 2, 3, 4\}$ are fixed, the

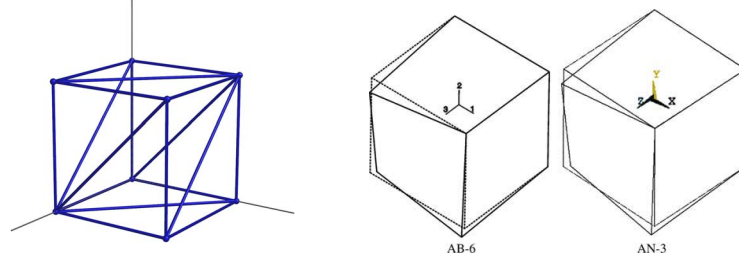


Figure 7.11: Residual distortion after 10 deformation cycles. While the result of our method does not show any visual residual (left) (numerical average residual displacement $1.1817E^{-11}$), the residual displacements for the inelastic elements ABAQUS-6 and ANSYS-3 are clearly visible (right). Right image courtesy by [Jabareen and Rubin, 2007].

nodes N_i , $i \in \{5, 6, 7\}$ are free, and node N_8 is moved along the path $A = (1.0, 1.0, 1.0)$, $B = (0.75, 1.0, 1.0)$, $C = (0.75, 0.75, 1.0)$, $D = (0.75, 0.75, 0.75)$, $E = (1.0, 0.75, 0.75)$, and $F = (1.0, 1.0, 0.75)$. As material parameters of our non-linear material model we choose an isotropic homogeneous material estimated from a block of a rubber-like material called Tango Black Plus. A detailed description of this material can be found in Section 8.5.

To quantify the residual distortion error, we performed 10 consecutive full deformation cycles along the path $ABCDEF A$. A resulting average residual displacement error based on the Euclidean distance of $1.1817E^{-11}$ indicates that our model shows an elastic response. This is an important property, Figure 7.11 compares the residual element distortion to elements ABAQUS-6 and ANSYS-3 and illustrates that small errors resulting in inelastic response are cumulative and could lead to significant visual artifacts.

The energy quantities (7.12) were computed by approximating the integral in Equation (7.14) using the trapezoidal rule, where each segment of the path was divided into 250 equal steps to ensure accuracy as suggested by [Jabareen and Rubin, 2007]. Although the resulting relative error values of the work are below 1% ($E_A^{\leftrightarrow} 0.49\%$, $E_A^{\leftarrow} 0.28\%$, and $E_D 0.29\%$), the results clearly suggest that the response of our method is Cauchy elastic. Further comparisons and error numbers are shown in Table 7.1.

Element	$E_A^{\rightarrow}(\%)$	$E_A^{\leftarrow}(\%)$	$E_D(\%)$	Type
Cosserat	8.3E-5	2.1E-14	-8.3E-5	Hyper
ABAQUS-1	3.3E-5	-5.3E-9	3.3E-5	Hyper
ABAQUS-2	-1.99	-0.069	-1.92	Cauchy
ABAQUS-6	0.393	0.011	-3.18	Hypo
ANSYS-1	2.5E-5	2.6E-9	-2.5E-5	Hyper
ANSYS-3	0.393	0.011	-3.18	Hypo
ours	0.49	0.28	0.29	Cauchy

Table 7.1: Relative path-dependent errors. Comparison of errors in the work as reported in [Jabareen and Rubin, 2007] with our method.

7.5 Discussion

In this chapter, we have presented a novel data-driven method for modeling non-linear heterogeneous soft tissue. The major practical contribution of our work is the ability to model rich non-linear deformations in a very simple manner, without the complex task of carefully choosing material models and parameters. Instead, our data-driven method relies on a simple-to-build acquisition system, a novel representation of the material through spatially varying interpolation of fitted linear models, and a simple deformation synthesis method.

Our work suggests a highly innovative approach to non-linear material modeling, but it also suffers from limitations. Due to its formulation, our technique is currently limited to capturing elastic properties. A fully dynamic simulation of soft tissue would require capturing other properties such as viscosity and plasticity. One interesting conclusion of our work is that it is often possible to obtain compelling surface deformations with a volumetric meshing unaware of an object's actual volumetric structure. This is of course not valid for all situations. For example, our face model could be greatly enhanced with accurate lip contact and jaw motion models.

There are several aspects of our model that deserve further exploration. One of them is its ability for capturing anisotropic behavior. The underlying linear co-rotational material model that we use for representing deformation samples can only capture isotropic behavior, but deformation samples with the same total strain but in different directions will lead to anisotropic behavior. In other words, we locally model the material isotropic in strain space, yet strain-space interpolation of material parameters provides global anisotropic

behavior. In the following chapter (Chapter 8), we show how to extend our model to anisotropy using an orthotropic formulation.

Another aspect that deserves further analysis is the formulation of the quasi-static deformation problem. Given a certain strain, we employ a local linear co-rotational model to formulate a quasi-static deformation problem. However, our model is not strictly a local linearization, which means that the stiffness matrix of the quasi-static deformation problem does not employ correct force derivatives. At the same time, our linear model is more robust than a model obtained by local differentiation and avoids non-passive regimes.

Similar to other approaches, our parameter fitting algorithm is formulated as a minimization problem and may end up in a local minimum. In fact, we have identified fitting error as the major source of potential inaccuracies in the deformation synthesis. Sometimes, fitting error also appears because we limit Poisson's ratio to physically valid values during the minimization. Robust parameter identification is still an open research problem in materials science, and some recent approaches explore alternative solutions including particle filters [Burion et al., 2008]. Multi-resolution fitting may be another way of increasing robustness.

Currently, we only use measurements of forces in direction of the probe's shaft without explicit measurements of tangential forces and friction behavior. While previously determining an adequate sampling of the object is impossible because of the unknown internal structure and heterogeneity, using a more efficient parameter estimation algorithm for material fitting, one could evaluate the need for further samples of the stress-strain relationship online, and determine the optimal probing patterns on the fly.

In the next chapter, we show as a further application area and validation of the presented material model. We discuss its applicability and employ it for designing and fabricating materials with desired deformation behavior.

Design and Fabrication of Materials with Desired Deformation Behavior

In the previous chapter we have shown how convincing deformable models can be obtained by using a measurement-based representation and modeling technique for simulating non-linear heterogeneous soft tissue. The continuous range of spatially varying material parameters was constrained to be in a physically valid range and the coefficients were selected to match the data best. In this chapter, we go a step further and want to achieve a desired large-scale behavior through appropriate discrete combinations of a set of available materials at a smaller scale. We introduce a data-driven process for designing and fabricating materials with desired deformation behavior. Our process starts with measuring deformation properties of base materials. For each base material we acquire a set of example deformations using the measurement system previously described in Section 6.3 and represent it as a curve in stress-strain space (Chapter 7). For material design and fabrication, we introduce an optimization process that finds the best combination of stacked layers that meets a user's criteria specified by example deformations. Our algorithm employs a number of strategies to prune poor solutions from the combinatorial search space. We demonstrate the complete process by de-

signing and fabricating objects with complex heterogeneous materials using modern multi-material 3D printers.

8.1 Introduction

When we want to design elastic deformable objects, either in computer animation or in the real world, we are faced with the problem of determining material descriptions and parameters such that the objects behave in a desired way.

Deformation effects can be modeled at very diverse scales, ranging from molecular interactions to globally-supported response functions, and through continuum elasticity laws or lumped-parameter models [Zohdi and Wriggers, 2004]. As recently demonstrated by work in numerical coarsening and homogenization, the behavior of materials with microscale inhomogeneities can be approximated by mesoscale homogeneous materials [Kharevych et al., 2009]. We are interested in a process that can be regarded as the inverse of homogenization, i.e., we want to achieve a desired large-scale behavior through appropriate combination of materials at a smaller scale. Our approach fits in the category of goal-based design of deformable models, and it addresses the challenges of anisotropic, inhomogeneous, and non-linear behavior. Furthermore, it goes a step beyond computer animation and enables the physical fabrication of deformable materials with a desired behavior.

Our work is motivated by the recent development of multi-material 3D printers such as the OBJET Connex series [OBJET, 2009]. These printers are capable of manufacturing a variety of soft and hard materials with complex internal structures, making it suddenly possible to fabricate complex 3D objects with aggregate materials quickly, inexpensively, and accurately. This development offers new opportunities and challenges for physics-based animation research. Computer graphics has already contributed systems for designing and fabricating clothes [Okabe et al., 1992], plush objects [Mori and Igarashi, 2007], paper craft objects [Mitani and Suzuki, 2004], or surface microgeometry [Weyrich et al., 2009]. But there is a lack of tools and algorithms for designing, editing, and fabricating user-specified deformable objects.

In this Chapter we make three main contributions:

- ▶ An algorithm and representation for coarsening deformable models with microscale inhomogeneous behavior. Our solution uses a

data-driven approach and is capable of capturing non-linear behavior while maintaining runtime efficiency. Conceptually, this can be seen as an extension of the linear numerical coarsening approach by [Kharevych et al., 2009] to non-linear material behavior.

- ▶ A goal-based material design approach that approximates a desired mesoscale deformation behavior by microscale materials through combinatorial optimization. This is an inverse modeling approach, inverting the concept of homogenization.
- ▶ A *complete* reproduction process for deformable materials, including acquisition, fitting, efficient simulation, goal-based design, and fabrication.

The result of our design process serves as input to a 3D multi-material printer for the actual physical fabrication of deformable objects.

Our approach to measure, design, and fabricate materials with desired deformation behavior has a number of distinct steps, summarized in Figure 8.1.

We have collected a database of base materials, fabricated using a Connex 500 multi-material 3D printer, but also a variety of standard foams, gels, and rubbers purchased from the McMaster-Carr catalogue. These materials span a wide gamut of different deformations: from very soft to very hard and rigid (Section 8.6). We automatically measure deformations of these base materials subject to different forces using a robotic system (Section 6.3).

Next, we apply our coarsening algorithm to model the base materials using a data-driven non-linear stress-strain relationship in a Finite Element Method (FEM) (Section 8.2 and 8.3). This compact representation allows us to predict deformations of thicker or thinner versions of the base material samples. More importantly, we show that we can accurately predict deformations of arbitrary combinations of stacked base materials.

As the last step, we design composite materials that best match a desired deformation behavior using our combinatorial optimization algorithm (Section 8.4). In order to simplify the material design process, we introduce a goal-based optimization approach. The user specifies a material by providing example deformations and their corresponding forces, and our algorithm automatically computes the best-matching composite material. Because the configuration space is combinatorial and exponentially large, we use an efficient search strategy that prunes away states that yield poor matches to the desired material specifications.

We validate the simulation and material model by fabricating a number of different composite materials, measuring their deformations subject to a va-

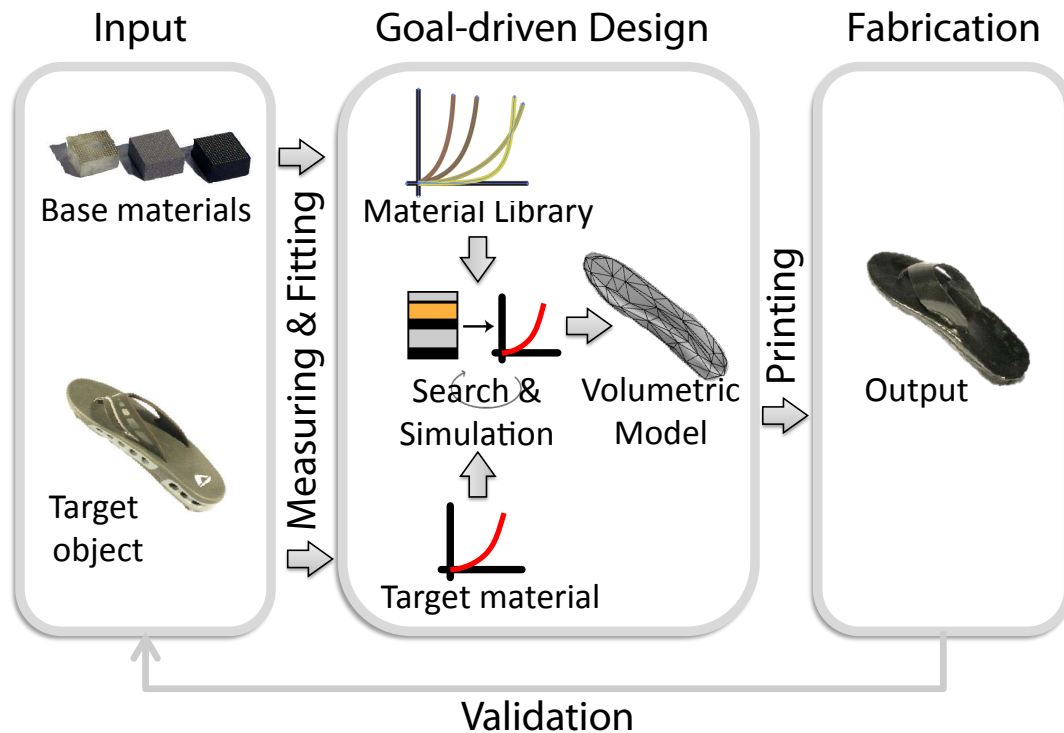


Figure 8.1: *Designing deformable materials. Our process starts with measuring deformation properties of base materials and the target object we want to duplicate. An optimization process finds the best combination of base materials that meets criteria specified by example deformations. Using a multi-material printer, we fabricate the real-world object with desired deformation behavior.*

riety of different forces and comparing these measurements to the results of the simulation. The results are described in Section 8.6.

8.2 Non-Linear Material Model

All our base materials exhibit a non-linear hyper-elastic stress-strain behaviour, as demonstrated by the measured force-displacement curves in Figure 8.3. Most of the base materials consist of complex structures on a microscale level which influences the deformation behavior significantly (see Figure 8.4). We use a data-driven approach and capture such non-linear behavior by a non-linear interpolation of locally linear material properties obtained from example deformations of the base material at a mesoscale. Con-

ceptually, this can be seen as a data-driven coarsening approach, representing the deformation behavior induced by microstructures on a mesoscale level.

To achieve this, we employ the concept of sampling the stress-strain function of the material at a small set of strain values as introduced in Chapter 7, and represent each stress-strain point by a linear co-rotational model. In the deformed state at a given point in the material with an arbitrary strain value, we represent the material locally by a non-linear interpolation of the sampled linear models.

Our linear FEM relies on Cauchy's linear strain tensor. Given a displacement field \mathbf{u} , it is defined as $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$. We obtain invariance of the strain under rotations by extracting the rotational part of the deformation gradient through polar decomposition, and then warping the stiffness matrix [Müller and Gross, 2004].

Thanks to symmetry of the strain and stress tensors, we can represent both as 6-vectors. Given the strain tensor, we construct its equivalent 6-vector as $\varepsilon = (\varepsilon_{xx} \ \varepsilon_{yy} \ \varepsilon_{zz} \ \varepsilon_{xy} \ \varepsilon_{xz} \ \varepsilon_{yz})^T$, and similarly for the stress. Then, a linear material can be represented by a 6×6 matrix E that relates the stress and strain vectors, $\sigma(\mathbf{u}) = E \cdot \varepsilon(\mathbf{u})$.

The key for achieving the non-linear behavior of the base materials is to define the stress-strain relationship, E , as a function of the local strain $\varepsilon(\mathbf{u})$. The matrix E can be typically parameterized by a smaller set of parameters \mathbf{p} . We define such parameters as a non-linear function of strain, therefore, the matrix E can be represented as a non-linear function $E(\mathbf{p}(\mathbf{u}))$. Although this approach holds for general anisotropic behavior, we describe subsequently the parameters \mathbf{p} for two types of materials that are most relevant in practice: isotropic and transversely isotropic ones. The transversely isotropic material will be further discussed in Section 6.3.

Isotropic Materials For homogeneous linear isotropic materials, the matrix E can be represented by the two Lamé's parameters λ and μ . In other words, the parameter vector is $\mathbf{p} = (\lambda, \mu)$. Using Lamé's parameters, the stress and strain tensors are related as

$$\sigma(\mathbf{u}) = 2\mu\varepsilon(\mathbf{u}) + \lambda \text{tr}(\varepsilon(\mathbf{u}))I, \quad (8.1)$$

from which the matrix E can be derived [Bathe, 1995].

In homogeneous linear isotropic materials, the strain can be well captured by the three invariants of the symmetric strain tensor $I_1(\varepsilon)$, $I_2(\varepsilon)$, $I_3(\varepsilon)$. These invariants do not change under rotation of the coordinate system. Using the

invariants to represent the strain, our non-linear material model can be considered as a 2-valued function in a 3-dimensional domain, $\mathbf{p}(I_1, I_2, I_3) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$.

Non-Linear Interpolation of Material Properties Given a base material, we describe its non-linear stress-strain relationship through a small set of P parameter vectors, $\{p_i\}$, corresponding to different strain values, $\{\varepsilon_i\}$. Then, using the (parameter, strain) pairs as RBF centers, we define the complete material behavior through RBF interpolation:

$$\mathbf{p}(\varepsilon) = \sum_{i=1}^P \mathbf{w}_i \cdot \varphi(\|\varepsilon - \varepsilon_i\|). \quad (8.2)$$

Since our base materials are homogeneous, a single set of parameter vectors is sufficient to describe the behavior of an arbitrary object consisting of a single base material. This reduces the number of parameters of a base material to $|\mathbf{p}| \cdot P$, where $|\mathbf{p}|$ is the cardinality of the parameter vector (2 for isotropic materials, and 5 for transversely isotropic ones). In our examples, the number of RBF centers is typically between $P = 6$ for the isotropic foams and $P = 12$ for printed materials with complex internal microstructure. Computing the RBF interpolation in a spatially varying manner, based on the local strain, allows us to simulate different non-linear behavior in different regions of an object.

In order to simulate the behavior of composite objects made of base materials, we follow the quasi-static FEM approach described in detail in Section 7.2. Given a simulation state, we compute the strain of all elements and perform a per-element computation of the parameter vector according to Equation (8.2). We then recompute the per-element stiffness matrices, and perform a new step of the FEM simulation.

8.3 Fitting Base Materials

We fit the properties of base materials such that simulated deformations match best a set of input examples. In the fitting process, we need to compute the RBF centers $\{\varepsilon_i\}$ (i.e., strain values used as data points), and their corresponding weights \mathbf{w}_i (See Equation (8.2)).

Let us first assume that the P RBF centers are known. Given a set of example deformations with measured displacements $\{\bar{\mathbf{x}}_i\}$ and corresponding forces

$\bar{\mathbf{F}}_i$, we compute the RBF weights \mathbf{w} by minimizing the error in the displacements, as:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \left\{ \sum_{i=1}^n \|\mathbf{x}_i(\mathbf{p}, \bar{\mathbf{F}}_i) - \bar{\mathbf{x}}_i\|^2 \right\}. \quad (8.3)$$

In order to define the RBF centers, we first fit the material as a homogeneous linear one and obtain a constant set of material parameters. Using these parameters, we run FEM simulations for all measurements, and record strain values. We select the RBF centers by sampling the strain space with P points that cover the range of measured values. Using these RBF centers we can fit the material parameters, but we run several iterations to obtain a better coverage of the strain space.

There are two main differences between this material fitting strategy and the one proposed in Section 7.3, motivated by prior knowledge about our base materials. First, since the base materials are homogeneous, the RBF weights are not spatially-varying, and the size of the problem is only $|\mathbf{p}| \cdot P$. Second, the objective function is defined by grouping the measured displacements of all example deformations at once. These two differences lead to improved robustness and fitting accuracy.

We use Levenberg-Marquardt optimization and compute the Jacobians as defined in Appendix B. An unconstrained optimization problem may lead to material parameters that are not physically correct. In the case of isotropic materials, it is easy to bound Lamé's parameters by computing the Young modulus and Poisson ratio and projecting these to physically valid values. In the case of transversely isotropic materials, we make sure that the stiffness matrix is positive definite using the technique of [Rebonato and Jäckel, 1999].

The measured forces $\bar{\mathbf{F}}$ are normal to the surface. However, the contact area below the force probe also undergoes small tangential forces during the measurement process, and we found that these missing forces produce small fitting errors. We increase the fitting quality by computing the missing tangential forces that would produce a perfect match at the probed surface points, and then reintroduce the tangential forces as known forces in the optimization problem.

Section 8.6 evaluates the quality of the fit and reports error values by comparing the simulation of fitted base materials to measured example deformations.

8.4 Goal-Based Material Design

Our main contribution is a goal-based material design approach that approximates a desired mesoscale deformation behavior by microscale materials through combinatorial optimization. We now describe the optimization algorithm to obtain composite structures made of a set of base materials. Our algorithm receives as input a description of the object surface, examples of desired force-displacement pairs, and a set of base materials with known deformation properties expressed in our non-linear material model.

8.4.1 Formulation of the Problem

We formulate the design process as an optimization problem where we need to choose the distribution of M possible base materials inside the fabricated object such that it matches the input force-displacement examples.

We discretize the problem by dividing the desired object shape in a set of N regular *cells*, each made of a uniform base material. The desired inhomogeneity and possible anisotropy of the final object are achieved by the appropriate distribution of base materials. At each cell, one may choose a single material from M possible base materials. We call a certain choice of base materials and their distribution a *design*. We denote each design as a vector $\mathbf{m} = (m_1, m_2, \dots, m_N)$, where m_i is an integer value that indicates the type of base material in the i^{th} cell out of the $\{\bar{m}_j, 1 \leq j \leq M\}$ possible base materials.

In order to test each design, we assign its particular material choices to the cells of the object, simulate the object with the quasi-static FEM approach from Section 8.2 using the user-specified force profiles, and measure the error in surface displacements. The surface displacements of all input examples are grouped in one large vector \mathbf{x} . Given the goal displacements $\bar{\mathbf{x}}$, the *displacement error* of a design is simply $\|\mathbf{x} - \bar{\mathbf{x}}\|$. Finding the optimal design with minimal displacement error is an exponential problem, with M^N possible designs to be tested.

8.4.2 Branch-and-Bound with Clustering

The major problem when solving such a design optimization problem is the non-convexity of the design space and therefore the risk of ending

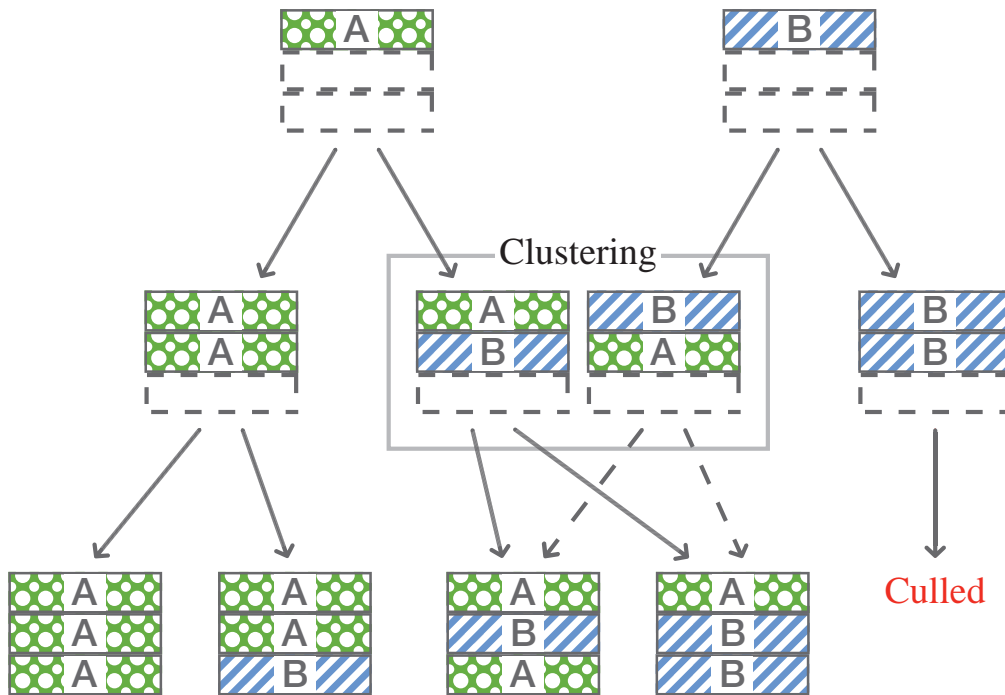


Figure 8.2: Branch-and-bound with clustering. The root of the tree shows the two materials A and B for the first out of three cells. Each level of the tree spans the possible options for the subsequent cells. Sub-optimal branches of the tree can be culled, and similar deformations can be clustered.

up with a local optimum solution if only the local neighborhood is observed [Lund and Stegmann, 2005].

To solve this discrete optimization problem we therefore use a decision tree, such that at each level of the tree we span the options for one cell in the design. The root of the tree has M children, where each child represents one of the material choices for the first cell, while the other $N - 1$ remain undecided. Figure 8.2 shows the decision tree for an object with three cells and two possible material choices.

Entire branches of the decision tree can be culled away using a branch-and-bound algorithm [Land and Doig, 1960]. During tree traversal, we store the minimum error d_{min} for the designs tested so far. When a new node of the tree is visited, i.e., a new cell is refined, we use this minimum error to cull (if possible) the complete subtree rooted at the node.

Given the breadth of the tree, branch culling still leads to an intractable num-

ber of possible designs to be tested. However, often several designs produce similar deformation results. We cluster these nodes together to limit the breadth of the decision tree at every level, and thus limiting the number of designs to be tested.

Bound Estimation Let us define as $\{\mathbf{m}\}_a = (m_1, m_2, \dots, m_l, x \dots x)$ the designs rooted at a node a and located at level l . The first l cells are already determined along this branch, while the rest are still undecided (denoted by x). We estimate a bound on the deformations produced by $\{\mathbf{m}\}_a$ by considering the cases where the undecided cells are uniform. In other words, to estimate bounds we fill the undecided cells with each one of the base materials \bar{m}_j , compute the resulting deformation for all input examples \mathbf{x}_j , and then bound the result of the M cases as $[\mathbf{x}_j]$. We use axis-aligned bounding boxes in high dimensions as bounds, i.e., maximum and minimum values for each dimension of the resulting displacement vectors. We cull the branch rooted at node a if $dist(\bar{\mathbf{x}}, [\mathbf{x}_j]) > d_{min}$. When new designs are tested we update d_{min} as appropriate.

The bound estimation approach is not conservative, due to material non-linearity and the existence of several non-monotonic functions along the simulation process. Recently, efficient methods for bounding displacements in linear FEM settings [Neumaier and Pownuk, 2007] were presented, but practical bounds for non-linear settings are still an open research challenge. However, the uniform blocks can be regarded as extreme behaviors (from very soft to very hard), therefore we expect that combinations of these materials will produce in-between deformations, in which case our bound estimation will not cull optimal designs.

Clustering Strategy We traverse the decision tree in a breadth-first manner, and hence a parent level with n nodes produces another level with $n \cdot M$ nodes. Evaluating bounds on this new level requires the computation of $n \cdot M^2$ designs. In order to limit the breadth of the tree, and thereby the total number of designs to be tested, we cluster nodes at every level before the splitting operation.

We cluster the n nodes at a level into K clusters using K-means clustering, using as distance $dist(a, b)$ between two nodes the sum of squared example dis-

placement differences evaluated for the pairwise uniform descendants. Formally, the distance metric is:

$$dist(a, b) = \sum_j^M \|\mathbf{x}(m_{1a}, \dots, m_{la}, \bar{m}_j, \dots, \bar{m}_j) - \mathbf{x}(m_{1b}, \dots, m_{lb}, \bar{m}_j, \dots, \bar{m}_j)\|^2. \quad (8.4)$$

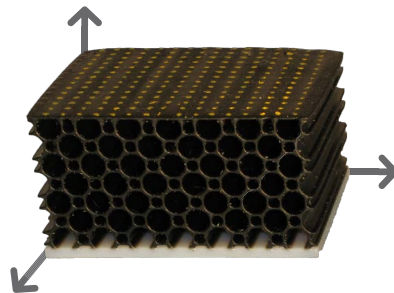
The cluster representative is the node that is closest to the centroid of the cluster. Every time we split a level we need to test only KM^2 designs. Since the height of the tree is equal to the number of cells N , our clustering strategy limits the total number of design evaluations to roughly $O(KM^2N)$. Note that the actual number of tested designs is actually smaller due to bound-based culling. In our implementation, we usually use $K = 20$ clusters. This clustering approach comes at the cost of missing the global optimal solution. However, we observed in our experiments that designs are usually close to the global optimum (see Section 8.6).

8.5 Base Materials

Isotropic Base Materials To print 3D deformable objects and a set of base materials we use the OBJET Connex 500 multi-material printer. In each run, the printer can use up to two different materials, e.g., Vero White (rigid) and Tango Black Plus (soft). The printer can also mix these two materials in pre-defined proportions producing isotropic materials of intermediate stiffness. We mainly use Tango Black Plus (TBP) and a mixed material called digital material with shore 50 (DM50). In addition to these two isotropic base materials we additionally measured eight isotropic materials from the McMaster-Carr online catalog, including rubbers and foams. Figure 8.3 shows a plot of surface displacement as a function of applied force for a subset of measured materials.

Transversely Isotropic Base Materials

In order to model and fabricate materials with even larger deformation gamut (in particular, materials that are much softer) we introduce internal empty spaces into the printed objects. Unfortunately, the current printer only allows printing empty spaces that span the entire object along the z-axis.



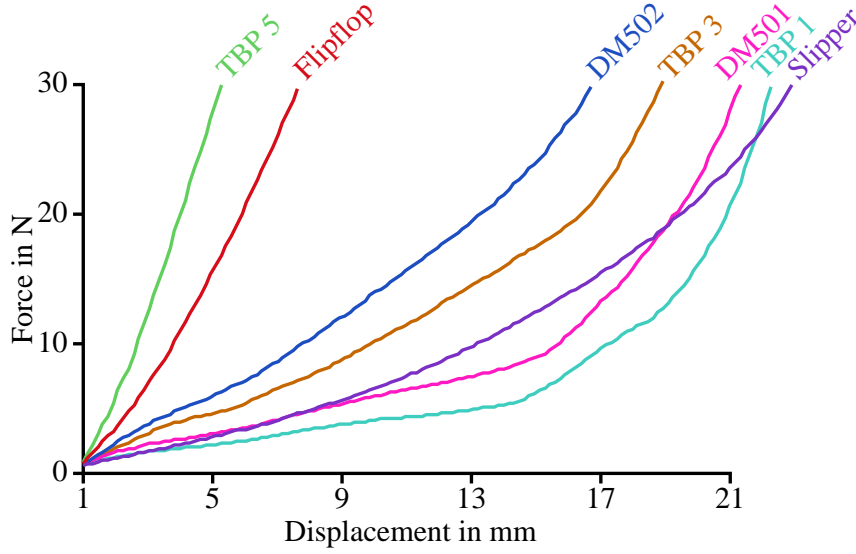


Figure 8.3: Force-displacement curves for a subset of the measured materials showing their non-linear behavior.

We use tubes of four different sizes and distributions (see example on the right). These objects are isotropic in the horizontal plane, perpendicular to the tube direction. The material can be regarded as *transversely isotropic*.

For such materials, the matrix E can be represented as:

$$E = \begin{bmatrix} E_{11} & E_{12} & E_{13} & 0 & 0 & 0 \\ E_{12} & E_{11} & E_{13} & 0 & 0 & 0 \\ E_{13} & E_{13} & E_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & E_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & E_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(E_{11}-E_{12})}{2} \end{bmatrix}. \quad (8.5)$$

with five degrees-of-freedom, $\{E_{11}, E_{12}, E_{13}, E_{33}, E_{44}\}$. Our non-linear material model can then be considered as a five-valued function in a six-dimensional strain domain, $\mathbf{p}(\boldsymbol{\varepsilon}) : \mathbb{R}^6 \rightarrow \mathbb{R}^5$.

All base materials were printed as 4 cm (width) \times 5 cm (length) \times 2.5 cm (height) blocks. The deformations (side view) of some of these materials under 15 Newtons force are shown in Figure 8.4. We will make the full set of base materials including all measurements and specifications public and plan to measure more materials in the future.

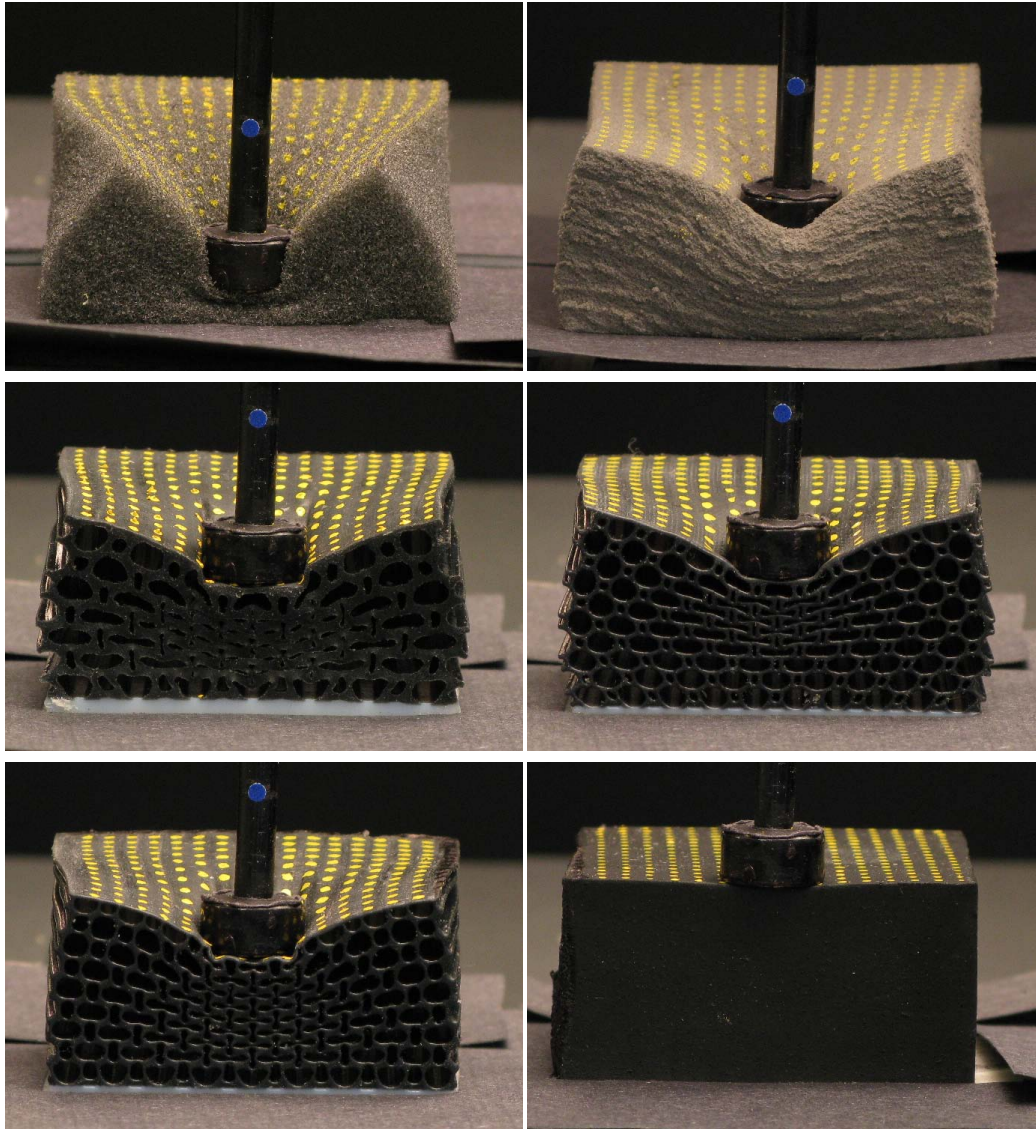


Figure 8.4: Pictures of base materials under 15 Newton force, showing the range of deformations under the same force load. Each block is of size 4 cm (width) \times 5 cm (length) \times 2.5 cm (height).

8.6 Validation and Results

Validation of the Fitting Our material model represents elastic behavior of the base materials at the mesoscopic level very well. In Figure 8.5 we compare images from our measurement system, the reconstructed deformed surface, and the corresponding simulation based on the FEM. We also show an error plot between the measured surface and the simulation. Note that the error is only evaluated at the surface marker positions and then interpolated for visualization purposes. Furthermore, the error evaluation is dependent on the accuracy of the measurement system which is in the range of < 1 mm. Very small pitching effects at the microscale of the material cannot be tracked by our system and are therefore missing in the error visualization. Refer to the video for more results. For isotropic base materials we use six and for the transversal isotropic base materials 12 RBF centers, resulting in 12 and 60 parameters for each base material, respectively. The fit to the material model takes two hours on average but has to be performed only once. We also report the average, standard deviation, and maximum errors for the materials under varying applied loads in Table 8.1.

Validation of the Stacking Next, we show that we can accurately predict the behavior of composite materials made from arbitrary combinations of base materials. We ran a number of simulations for different composites and also fabricated those using the Connex 500 printer. Next, we measured the behavior of these composite materials using our system and compared them to their corresponding simulations. We report this validation for a few example deformations and materials in Figure 8.6 and in the companion video. In the composite example shown in Figure 8.6, we obtain average errors of 1.98 mm and 2.16 mm under loads of 10 N and 20 N.

Validation of the Goal-based Design Next, we validate our goal-based design process. As the first step we tested it on materials that we know we can reproduce. We picked a given combination of layers and their thicknesses. Then we simulated this composite material and used its deformations as the input to the search algorithm. We report the result of this validation in Figure 8.8. We tested this strategy on 20 different randomly chosen material designs (5 layers, each with 9 different material choices and 5 force-displacement pairs). Although our search is not guaranteed to find the global optimum, it always found a very close solution (average RMS error of 0.067 mm). The optimization time is usually below one hour. To carry this valida-

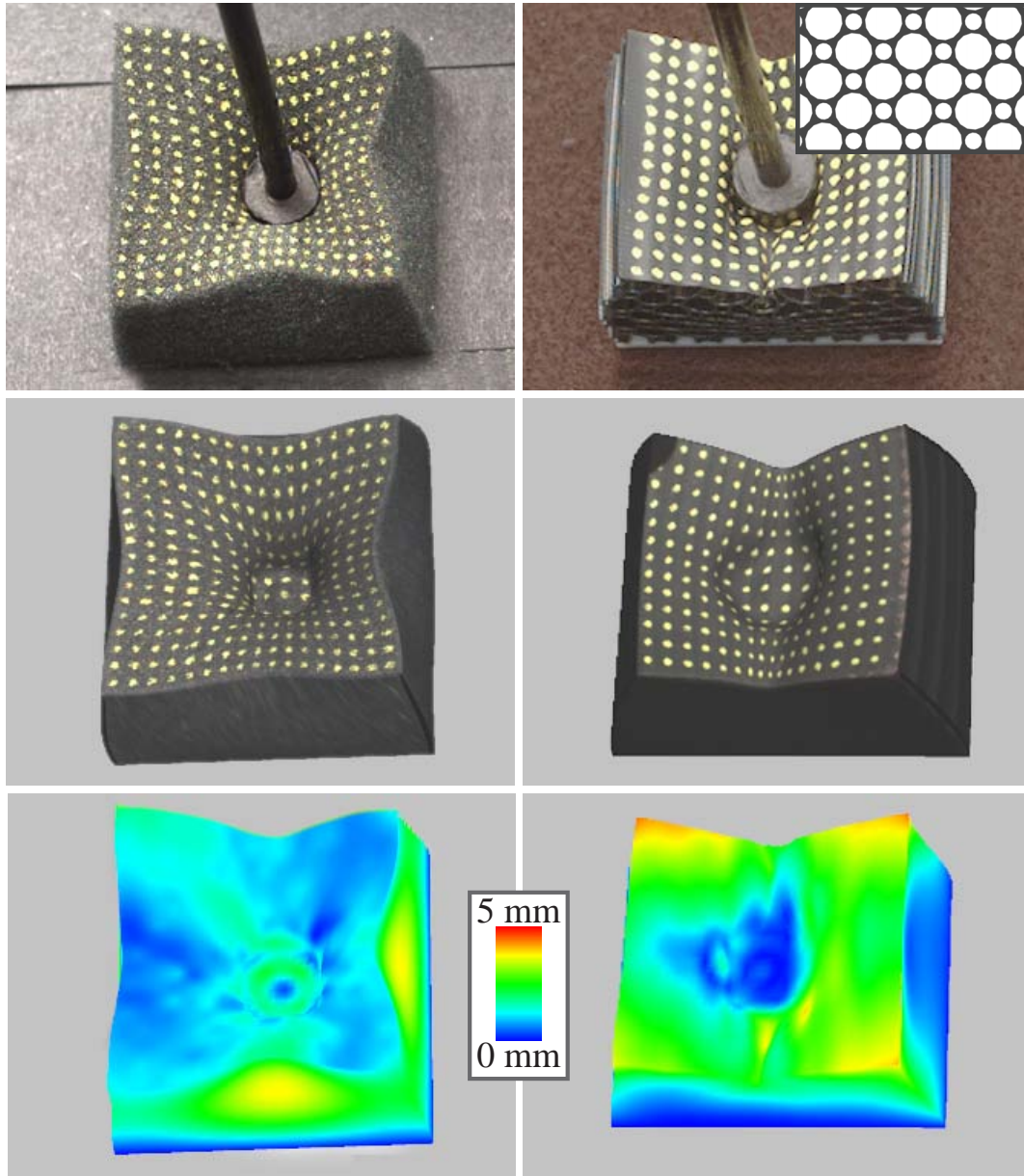


Figure 8.5: Side-by-side comparison of real and simulated materials. Deformation of an isotropic (left column) and transversely isotropic material (right column), comparing acquisition (top row) with the simulation (middle row) and the displacement error (bottom row).

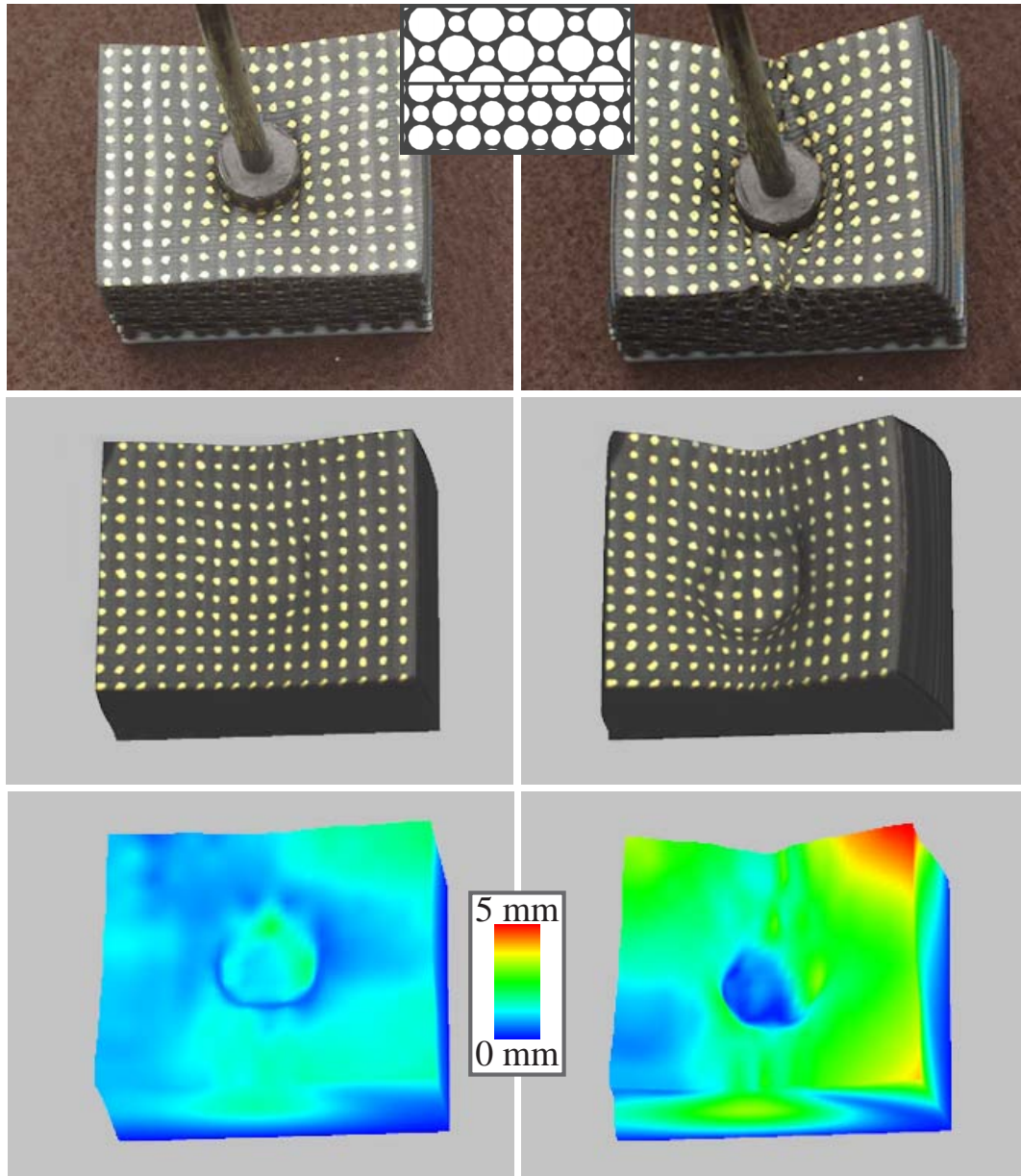


Figure 8.6: Validation of composite materials. We assigned the material properties obtained from two independent fits of base materials (DM501 and DM502) to a composite material consisting of two layers. We then printed the composite and compared the deformations of the real object (upper row) to the simulation (middle) under a load of 8 and 21 Newton. Lower row: Error visualization.

Material	Force (N)	Displacement error (mm)		
		avg.	std.dev	max
Foam (very soft)	1N	0.84	0.45	2.55
	3N	1.72	1.00	6.74
	5N	2.04	0.88	5.70
Foam (medium)	5N	1.40	0.57	3.44
	15N	1.06	0.36	2.46
	25N	1.33	0.90	5.19
Foam (stiff)	10N	0.73	0.43	2.70
	20N	0.94	0.40	2.71
	30N	1.20	0.38	2.22
Printed TBP1 (soft)	5N	2.14	0.68	4.47
	10N	2.40	0.77	4.67
	20N	3.60	1.22	6.55
Printed DM502 (medium)	5N	0.69	0.26	1.44
	15N	0.85	0.41	2.09
	25N	1.31	0.51	3.00
Printed TBP5 (stiff)	10N	0.68	0.22	1.15
	20N	0.99	0.27	1.70
	30N	1.30	0.311	2.61

Table 8.1: *Error evaluation of the model. We fitted parameters for various isotropic (soft/medium/hard foams) and transversely isotropic (printed materials with cylinder hole structures) materials and evaluated the surface displacement error under small, medium, and high force loads by comparing to measured deformations of material blocks (size isotropic 5x5x2.5cm, printed 5x4x2.5cm).*

tion even further, we have fabricated these composites and remeasured their properties. We show these results in Figure 8.8.

We also tried to approximate one of the foams with a combination of materials printed using the Connex 500. The obtained spatial combination and the error evaluation are shown in Figure 8.9.

Replicating objects For the most complex results, we try to showcase the whole process by replicating objects with complex deformation behavior. In particular, we reproduced a beach flip-flop, a felt slipper, and a heterogeneous leather stool. We used a Cyberware scanner to scan the 3D geometry of each of the objects. Then, we acquired the deformation behavior of each



Build resolution	
X-axis	42 μ m
Y-axis	42 μ m
Z-axis	16 μ m

Figure 8.7: *Objet Connex 500 3D multi-material printer. Image: Final result after approximately 10 hours printing. White material is support material which is removed using a high-pressure water jet. Table: Build resolution of the printer in high-quality mode [OBJET, 2009].*

object using our measurement system and fitted material parameters. For the leather stool, we segmented the volume into two areas, and approximated each of them as a homogeneous material. Next, we used the goal-based design process to find the best approximation of the material deformation properties using our base materials. For all results, we used between 5 and 10 force-displacement pairs. Finally, we printed replicas of these objects using the multi-material printer. As can be seen in Fig. 8.11 and 8.10 and the accompanying video the replicas show very similar behavior to the original objects. To further validate this approach we show force-displacement curves in Fig. 8.11 and 8.10 for the corresponding points on the replicas and the original objects.

8.7 Discussion

We have presented a complete approach for measuring, designing, and fabricating materials with desired deformation behavior. Our model is able to represent and simulate the non-linear elastic deformation behavior of objects with complex internal microstructure. In order to ensure a good match between deformations of real materials and their simulated behavior, we use a data-driven measurement process to estimate non-linear stress-strain models

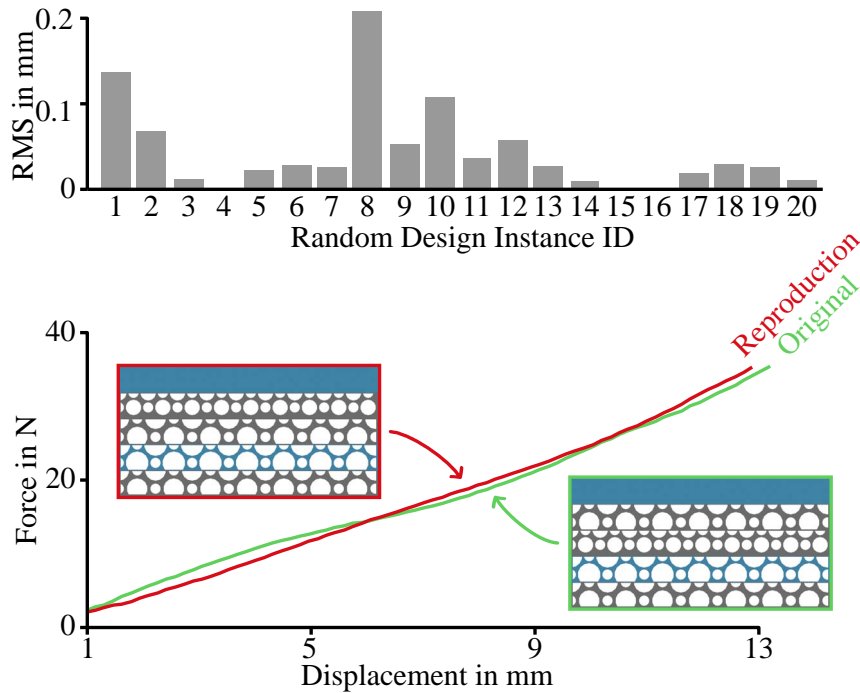


Figure 8.8: Validation of goal-based design algorithm. We randomly generate a set of material designs. We then simulate these designs and use their simulated deformations as the input to the goal-based design search algorithm. We then compare the obtained designs of the search algorithm with the known ground truth. The upper bar plot shows the RMS error. We also fabricated one of those randomly generated designs and its corresponding search output and compared their force-displacement curves.

for each material. Furthermore, we show that a goal-based material design approach can approximate a desired global deformation behavior by finer scale materials through combinatorial optimization. By closing the loop between measurements, simulation, goal-based material design, and printing, we validate the complete pipeline and show that close matches between simulated and real fabricated objects can be achieved. Our goal-based design is a significant step towards 3D hardcopying.

Limitations and Future Work. We believe that our system has many potential avenues for improvements and future work. We predict that this process will be a template for many future systems that expand the range of simulated and fabricated material properties (such as dynamic deformation properties or plasticity). More specifically, we plan to extend our model to dynamic and plastic deformation behavior and improve our measurement

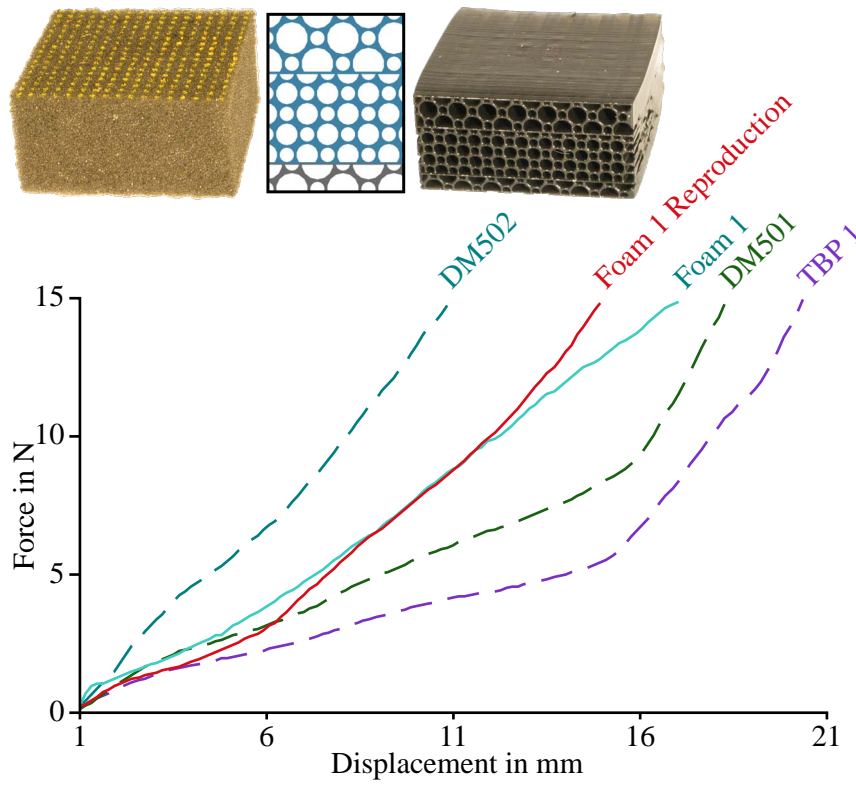


Figure 8.9: Fabricated example of goal-based design. As an input to our goal-based material design algorithm, we specified 5 example deformations of a foam (upper left). The desired deformation behavior is approximated by finer scale materials obtained through combinatorial optimization and then fabricated using a 3D printer (upper right). The lower curve shows the force-displacement relationship of used base materials, foam, and fabricated approximation.

system such that it can acquire a wider range of deformation properties (e.g., material stretching and dynamic deformation measurements) or can guarantee and incorporate prior physical knowledge, such as volume preservation. Additionally, we plan to investigate strategies for optimally choosing the number of degrees of freedom (RBF centers) of our material model, striking a balance between accuracy and overfitting. Furthermore, we would like to examine material homogenization strategies to improve the speed of the forward (simulation) step for non-linear materials. This improvement along with more advanced search strategies could, in turn, speed up the inverse step such that we could design and fabricate extremely complex heterogeneous materials.

Spatial combinations. Currently, we only print layers of different materials. However, we believe our algorithm could be extended in a straight forward manner to arbitrary spatial combinations (e.g., voxels) of base materials. The decision tree could be directly applied to 2D or 3D problems, by having a one-to-one mapping of layers in 1D to voxels in 3D. Also, our pruning strategy (clustering and bounds) can be directly translated to the 3D case. Our search algorithm linearly scales with the number of layers or volume elements.

For current printers, the mechanical range of isotropic base materials without any holes or tube structure is limited. The OBJET Connex 500 printer can mix two different materials, and the material properties are restricted to the range between the two base materials. To significantly expand this range, we deliberately decided to create tube-structured materials. Due to current physical printer limitations, these void tube structures can only be printed along the z-axis of the printer, otherwise they would get filled with structure material, which is difficult to remove. Printing blocks or objects with isotropic hole structures (similar to Swiss cheese) is currently not possible. This comes at the cost of requiring a transversely isotropic material model.

Looking into the future, we predict that the next generation of 3D multi-material printers will be able to use many more base materials with a wider range of material properties and more complex internal structures. As the cost of these printers decreases and their capabilities increase, we believe that the goal of personalized design, modeling, simulation, and fabrication will become reality.

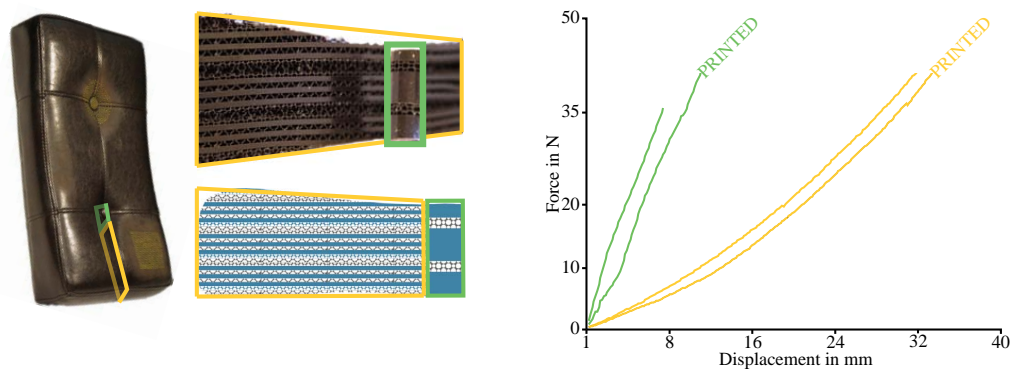


Figure 8.10: *Replicating a leather stool. The left column shows the original object, the middle column a cross section of the replicated object and the spatial combination of base materials. We segmented the stool into two regions, a stiff region below the button (indicated in green) and the remaining softer region (indicated in orange). We validated the deformation behaviour by comparing the force displacement plots (right column) in the button region (orange) as well as in the softer region (green).*

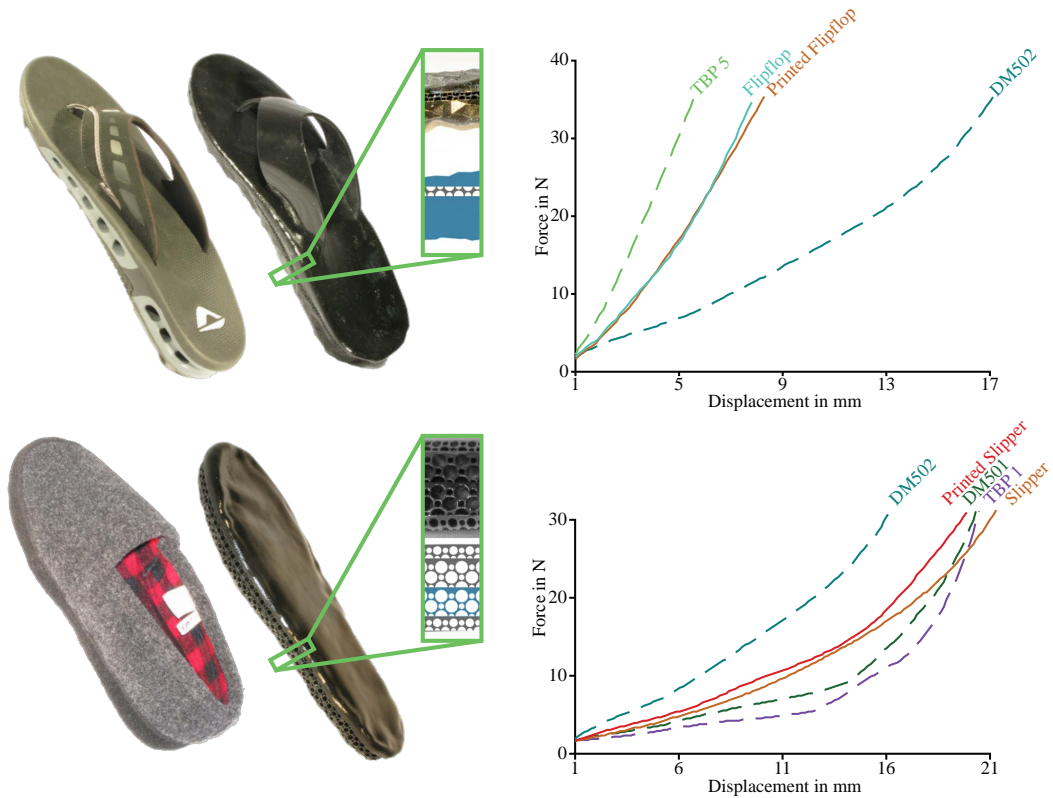


Figure 8.11: *Replicating a flip flop (upper row) and a slipper (lower row). The left column shows the original images, the middle column the replicated flop flop and slipper with the spatial combination of base materials obtained by our goal-based optimization approach. Our replication matches the deformation behaviour of the original quite well, as shown in the force displacement plots (right column) for a corresponding point on the original and replica. The dotted curves characterize the base materials.*

Conclusion

This chapter concludes the thesis by summarizing and discussing the major contributions and suggesting future research directions.

9.1 Discussion

This thesis presented a combination of data-driven and physically based techniques for acquiring, modeling, and animating deformable objects, with a special focus on human faces. Capturing of a small but representative amount of real data, and then synthesizing diverse on-demand examples with procedural or physically-based models and real data as input benefits from both sides: Highly realistic model behavior due to real-world data and controllability due to physically-based models.

In this context, we presented a three-dimensional dynamic face model that can accurately represent the different types of spatial and motion scales that are relevant for wrinkle modeling and animation. A central design element of our model is a decomposition of the facial features into fine, medium, and coarse spatial scales, each representing a different level of motion detail. This decomposition allows us to uniquely tailor the acquisition process to the spatial and temporal scale of expression wrinkle motions.

9 Conclusion

We acquire a static high-resolution model of the face, including reflectance data. Then we place markers on the face and mark expression wrinkles with a diffuse color. We add two synchronized cameras to a marker-based optical motion-capture system and capture the facial performance. We adapt a linearized thin shell model to deform the high-resolution face mesh according to the captured motion markers. From the video data we estimate the expression wrinkles using a 2D parametric wrinkle model and add them to the deformed 3D face mesh by solving a non-linear energy minimization problem. While the marker color limits the resolution of the system, in theory our system could be combined and benefit from very recent advances in camera sensors and passive shape acquisition systems [Beeler et al., 2010].

Based on this model we presented a *hybrid animation* technique that combines computational and data-driven approaches, and thereby achieves detailed facial expressions that are intuitive to control as well as efficient to compute. Building on the previously proposed multi-scale decomposition, we compute the large-scale motion using the same linear shell deformation, but in contrast employ a data-driven approach for learning fine-scale facial details from a small set of example poses. In a preprocess we learn the correlation of wrinkle formation to sparsely measured skin strain. At run-time, given an arbitrary facial expression, it computes the skin strain and derives from it the necessary fine-scale corrections for the large-scale deformation. The resulting model is compact, runs in real-time on a GPU, and allows editing operations such as transferring wrinkles onto face models that lack of facial details.

Following the the concept data-driven interpolation in strain space, we presented an algorithm that transfers measured example deformation into a local element-wise strain space, and represents this example deformation as a locally linear sample of the material's stress-strain relation. The full non-linear behavior is modeled by interpolating the material samples in strain space using radial basis functions (RBFs). Finally, a simple elastostatic finite-element simulation of the non-linearly interpolated material samples based on incremental loading allows for efficient computation of rich non-linear soft-tissue simulations. The method significantly simplifies the construction of convincing models based on a set of measured example deformations of real-world objects, thereby avoiding complex selection of material parameters.

Based on this method, we introduced a data-driven process for designing and fabricating materials with desired deformation behavior. Our goal-based material design approach approximates a desired mesoscale deformation behavior by microscale materials through combinatorial optimization. This is an inverse modeling approach, inverting the concept of homogenization. We

demonstrated a complete reproduction process for deformable materials, including acquisition, fitting, efficient simulation, goal-based design, and fabrication. The result of our design process serves as input to a 3D multi-material printer for the actual physical fabrication of deformable objects.

9.2 Future Work

There are several aspects in the area of hybrid deformation models that are not covered by this thesis.

While we presented efficient tools to capture a small but representative amount of real data, and then synthesizing diverse on-demand examples with physically-based methods, there is currently a lack of knowledge on proper sampling and data-acquisition planning. An important research area would be to investigate subspaces and perceptual aspects of facial expressiveness, motion, and appearance. Insights in these areas would allow for example tailoring data-acquisition processes, and developing faster, smaller, lightweight, and highly portable capture devices.

Furthermore, this could leverage artistic control to a higher level. In our opinion, one of the biggest research challenge will be mapping high level controls such as emotions and expressive speech to soft tissue movement and deformation. Solving this problem probably requires interdisciplinary cooperations in the area of computer vision, computer graphics, machine learning, bio-mechanics, and perceptual research.

To achieve both realistic and controllable models, an interesting approach would be combining statistical and anatomical models. As a specific example, a statistical model for wrinkles [Golovinskiy et al., 2006] extended to dynamics and combined with an anatomical model could guarantee physical correctness and provide high-level artistic control.

Anatomical accuracy would also require the addition of other important facial features like hair, teeth, or the eyes, which are outside the scope of skin deformation, but play a key role in the realism of the full face animation.

In the last two decades, in computer graphics a large number of deformation techniques, ranging from simple geometric models, variational surface deformation techniques, and mass-spring systems to linear and non-linear finite element models for a wide range of simulation purposes have been presented. However, acquiring and estimating suitable material parameters remains still very challenging and unintuitive. In this thesis, we only investigated non-linear elastostatic behavior of soft tissue. Extending the model

9 Conclusion

to dynamic simulations and plasticity while guaranteeing simple parameter acquisition would be an interesting research challenge.

Another interesting field that has not been addressed in this thesis are user interfaces and artistic design processes for modeling deformable objects. Interactive tools visualizing the range of physically plausible deformations and material configurations available by output devices such as rapid prototyping machines and haptic feedback could be valuable tools for designers.

Material optimization strategies in general have a wide range of applications, such as designing real objects, coarsening and speeding up virtual physical simulations, or adding priors to material estimation strategies. In our case, a branch-and-bound search strategy in combination with physical bounds turned out to be efficient for optimizing elastic deformation behavior. Extending this to more general objective functions including additional factors such as material cost, weight, or even appearance, as well as tighter physical bounds and more general optimization strategies would be an interesting research challenge.

Looking into the future, we predict that hybrid deformation models, unifying data-driven and physically-based approaches, will become essential for realistic simulation behavior. Pure data-driven techniques usually require exhaustive data-acquisition, lack of extrapolation capabilities, and are difficult to edit. Physical methods are usually challenging due to model design and parameter estimation.

Capturing only a small but representative amount of real data, and then synthesizing diverse on-demand examples with physically-based models and real data as input benefits from both sides: Highly realistic model behavior due to real-world data and controllability due to physically-based models.

As the cost and complexity of acquisition systems, simulation methods, and rapid prototyping technologies decreases and their capabilities increase, we believe that personalized design, modeling, simulation, and even fabrication will become reality.

A P P E N D I X

A

Notation And Glossary

A.1 Operators

\cdot	Dot product.
\times	Cross product.
\otimes	Tensor product.
$\ \ \ $	Euclidean distance.
$\frac{\partial f}{\partial x}$	Partial derivative of scalar-valued function f .
$\frac{\partial \mathbf{f}}{\partial x}$	Partial derivative of vector-valued function \mathbf{f} .

A.2 Notation

\mathbb{R}	The set of real numbers.
\mathbb{R}^3	Three-dimensional space.
\mathbf{u}	Displacements.
t	Time.

Facial geometry and motion

H	Set of handle vertices or motion capture marker positions.
n	Number of motion capture markers.
$\mathbf{m}_{i,t} \in \mathbb{R}^3$	Time-dependent marker position in the reference space of the motion-capture system.
F	Face mesh.
$\mathbf{f}_{i,t} \in \mathbb{R}^3$	Time-dependent marker position in the reference space of the face space.
$\mathbf{u}_{i,t} = (\mathbf{f}_{i,t} - \mathbf{f}_{i,0}) \in \mathbb{R}^3$	Time-dependent marker displacements.
$\mathbf{c} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$	Correspondence warp function.
$\phi : \mathbb{R} \rightarrow \mathbb{R}$	Scalar basis function.
$\mathbf{p} : \mathbb{R}^3 \rightarrow \mathbb{R}$	Quadratic trivariate polynomial.
$\mathbf{d}_t : F \rightarrow \mathbb{R}^3$	Smooth deformation function.
Δ	Discrete Laplace Beltrami operator.
$\mathbf{v} : \mathbb{R} \rightarrow \mathbb{R}^2$	B-Spline curve.
$S(p) : \mathbb{R}^2 \rightarrow \mathbb{R}$	Wrinkle cross-section function.
$V(p, \omega)$	Visibility function, 1 if $(p, S(p))$ is visible from direction ω and 0 otherwise.
$I(p)$	Intensity at point p .
$\ e_i\ $	Length of the i 'th edge.
θ_i	Dihedral angle of the i 'th edge.
$\ t_i\ $	Area of the i 'th triangle.
$\mathbf{f} = [f_1, \dots, f_F]^T$	F -dimensional feature vector.
f_i	Relative stretch of the i 'th feature edge.
$\ \mathbf{f} - \mathbf{f}_j\ _v$	Weighted distance metric per vertex v .

Solid soft tissue modeling

\mathbf{u}	Displacement field.
$\varepsilon(\mathbf{u})$	Cauchy's linear strain tensor.
$\sigma(\mathbf{u})$	Stress tensor.
\mathbf{E}	Material tensor.
E	Young's modulus.
μ	Poisson ratio.
λ, α	Lamé's parameters.
K_e	Per element stiffness matrix.
R_e	Element's rotation matrix.
\mathbf{p}	Material parameter vector.
$\mathbf{x}_i(\mathbf{p}, \bar{\mathbf{F}})$	Simulated position of a mesh node as a function of material parameters and the measured forces.
$\bar{\mathbf{x}}_i$	Measured position of a mesh node.
$I_1(\varepsilon), I_2(\varepsilon), I_3(\varepsilon)$	Invariants of the symmetric strain tensor.
N	Number of cells.
M	Number of base materials.
m_i	An integer value that indicates the type of base material in the i^{th} cell out of the $\{\bar{m}_j, 1 \leq j \leq M\}$ possible base materials.
$\mathbf{m} = (m_1, m_2, \dots, m_N)$	A design (material configuration).
$\{\mathbf{m}\}_a = (m_1, \dots, m_l, x \dots x)$	The designs rooted at a node a and located at level l . The first l cells are already determined along this branch, while the rest are still undecided (denoted by x).
$dist(a, b)$	Sum of squared example displacement differences between two nodes evaluated for the pairwise uniform descendants.
K	Number of clusters.

A.3 Glossary

PCA	Principal Component Analysis.
SSD	Skeletal Subspace Deformation.
PSD	Pose Space Deformation.
RBF	Radial Basis Function.
SVM	Support Vector Machine.
WPSD	Weighted Pose Space Deformation.

Jacobian for Parameter Fitting

During fitting of material parameters $\mathbf{p} = \{\lambda_e, \alpha_e\}$ through minimization of Equation (7.11), we need to compute the Jacobian of the deformed vertex positions w.r.t. the parameters, i.e., $J = \frac{\partial \mathbf{x}}{\partial \mathbf{p}}$, in each iteration of the Levenberg-Marquardt algorithm.

Given external forces \mathbf{F} and initial positions \mathbf{x}_0 , the deformed positions under the linear co-rotational elastostatic problem [Müller and Gross, 2004] are

$$\mathbf{x} = K^{-1} (\mathbf{F} + K' \mathbf{x}_0), \quad (\text{B.1})$$

$$\text{with } K = \sum_e [R_e K_e R_e^T]_e \text{ and } K' = \sum_e [R_e K_e]_e.$$

Here $[\dots]_e$ denotes the assembly of the submatrix of the e -th element into the complete stiffness matrix.

The Jacobian w.r.t. each parameter $p_i \in \{\lambda_e, \alpha_e\}$ can then be computed as

$$J_i = \frac{\partial K^{-1}}{\partial p_i} (\mathbf{F} + K' \mathbf{x}_0) + K^{-1} \frac{\partial K'}{\partial p_i} \mathbf{x}_0, \quad (\text{B.2})$$

$$\text{with } \frac{\partial K^{-1}}{\partial p_i} = -K^{-1} \frac{\partial K}{\partial p_i} K^{-1}.$$

B Jacobian for Parameter Fitting

Note that we do not compute the inverse of K . Instead, we compute a sparse Cholesky factorization [Toledo et al., 2003], and then use this factorization many times for solving the linear systems above.

Recall the expression for the (unwarped) per-element stiffness matrix in Equation (7.9). The remaining terms are defined as:

$$\begin{aligned}\frac{\partial K}{\partial \lambda_e} &= \left[V_e R_e B_e^T G B_e R_e^T \right]_e, & \frac{\partial K'}{\partial \lambda_e} &= \left[V_e R_e B_e^T G B_e \right]_e, \\ \frac{\partial K}{\partial \alpha_e} &= \left[V_e R_e B_e^T H B_e R_e^T \right]_e, & \frac{\partial K'}{\partial \alpha_e} &= \left[V_e R_e B_e^T H B_e \right]_e.\end{aligned}\quad (\text{B.3})$$

In the case when some nodes are constrained not to deform (e.g., when the bottom of the captured objects is fixed), their known positions move to the right-hand side in Equation (B.1), and the Jacobians must be slightly modified.

A P P E N D I X

C

Curriculum Vitae

C Curriculum Vitae

Personal Information

First Name: Bernd
Last Name: Bickel
Address: Disney Research, Zurich
Clausiusstrasse 49
8092 Zurich, Switzerland
Phone: +41 762 130306
Email: bernd@disneyreserach.com
Homepage: <http://graphics.ethz.ch/~bickelb/>
Data and place of birth: June 03 1982 in Feldkirch, Austria

Education

Sept. 2001 - March 2006 Study of Computer Science at the Swiss Federal Institute of Technology (ETH).
July 2005 - March 2006 Internship Mitsubishi Electric Research Laboratories (MERL), Boston, MA.
since May 2006 Ph.D. student at the Swiss Federal Institute of Technology (ETH).
since July 2009 Researcher at Disney Research, Zurich.
Dec 2009 - Jan 2010 Research visit Graphics, Vision and Interaction Group at the School of Engineering and Applied Sciences at Harvard University.

Employment

July 2000 - Aug 2001 Internship Marketing Communication (part-time), OMICRON electronics GmbH, Klaus, Austria.
July 2005 - March 2006 Internship Mitsubishi Electric Research Laboratories (MERL), Boston, MA, USA.
May 2006 - Jun 2009 Research Assistant at ETH Zurich, Zurich, Switzerland.
since July 2009 Researcher at Disney Research, Zurich, Switzerland.

Publications

- SIGGRAPH 2010 B. Bickel, M. Bacher, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, W. Matusik: *Design and Fabrication of Materials with Desired Deformation Behavior*
- SIGGRAPH 2010 T. Beeler, B. Bickel, P. Beardsley, B. Sumner, M. Gross: *High-Quality Single-Shot Capture of Facial Geometry*
- SCIENCE CHINA Information Sciences 2010 H. Kim, B. Bickel, M. Gross, S. Choi: *Subsurface scattering using splat-based diffusion in point-based rendering*
- SIGGRAPH 2009 B. Bickel, M. Baecher, M. Otaduy, W. Matusik, H. Pfister, M. Gross: *Capture and Modeling of Non-Linear Heterogeneous Soft Tissue*
- SCA 2008 B. Bickel, M. Lang, M. Botsch, M. Otaduy, M. Gross: *Pose-Space Animation and Transfer of Facial Details*
- EUROGRAPHICS 2007 N. Pietroni, M. Otaduy, B. Bickel, F. Ganovelli, M. Gross: *Texturing Internal Surfaces from a Few Cross-Sections*
- SIGGRAPH 2007 B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, M. Gross: *Multi-Scale Capture of Facial Geometry and Motion*
- VMV 2006 B. Bickel, M. Wicke, M. Gross: *Adaptive Simulation of Electrical Discharges*
- Sketch SIGGRAPH 2006 B. Bickel, T. Weyrich, W. Matusik, H. Pfister, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen, M. Gross: *Implementation Sketch: Processing and Editing of Faces using a Measurement-Based Skin Reflectance Model*
- SIGGRAPH 2006 T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen, M. Gross: *Analysis of Human Faces using a Measurement-Based Skin Reflectance Model*
- ETH Zurich, master thesis 2006 B. Bickel: *A Measurement-Based Skin Reflectance Model*

List of Figures

1.1	Uncanny valley.	2
1.2	Research challenges for mastering and understanding human faces.	4
3.1	Multi-scale face model processing pipeline	22
3.2	Animation of a high-resolution face.	24
3.3	The cotangent discretization.	26
3.4	Process of synthesizing medium-scale wrinkles.	28
3.5	Preprocess of removing wrinkles that already exist in the initial, relaxed-pose face scan.	29
4.1	Face-capturing setup, consisting of six cameras for marker tracking complemented by two cameras for wrinkle capture.	34
4.2	Face capture setup for static high-resolution geometry and appearance data.	35
4.3	Establishing correspondence.	37
4.4	Fitting of a B-spline curve and corresponding cross-section shape for a wrinkle.	39
4.5	Wrinkle cross-section function $S(w, d, p)$	40
4.6	Performance replay of captured video sequences.	44

List of Figures

4.7	Side-by-side comparisons of original photographs and reconstructed facial expressions.	44
4.8	Art-directable editing of wrinkles by scaling or 2D sketching in video frames.	45
5.1	Interactive Editing of Highly-Detailed Faces.	49
5.2	Hybrid face animation pipeline.	50
5.3	Wrinkle Formation and Skin Strain.	53
5.4	Color coded weight kernels.	55
5.5	L^2 Error Vs. Size of the Basis.	58
5.6	Comparison of methods such as blend shapes, local blend shapes, PSD, and WPSD to original input data.	58
5.7	Real-Time Face Animation Results from Mocap.	62
5.8	Wrinkle Editing on an Actress' Performance.	63
5.9	Fine-Scale Transfer to a manually modeled human and a cartoonish character.	64
6.1	Trinocular Stereo Vision System.	69
6.2	Contact probe with integrated force sensor including USB Interface Kit, Force Sensing Resistor, and Phidget Voltage Divider.	70
6.3	Measured facial deformations.	71
6.4	Automated system for measurement of material deformations consisting of cameras, a robot arm, and a force sensor.	73
7.1	Processing pipeline of acquiring and modeling non-linear quasi-static soft tissue behavior.	76
7.2	Non-linear deformations and incremental loading.	78
7.3	Examples of a deformed pillow with color-coded Young's modulus.	80
7.4	Comparison of captured and synthesized deformations for a foam block and examples of interactive deformations.	82
7.5	Comparing real and modeled deformations with a different contact probe shape than the one used in the data acquisition phase.	85
7.6	Comparison of our method vs. an average-fit linear corotational model.	87
7.7	Comparisons of captured and synthesized deformations for a heterogeneous non-linear pillow and interactive deformations of the model.	88
7.8	Approximate skull model, facial tissue, and embedded high-resolution triangle mesh.	89
7.9	Facial deformation examples.	89
7.10	A test for path-independence of our element formulation.	90

List of Figures

7.11	Residual distortion after 10 deformation cycles.	91
8.1	Overview designing deformable materials.	98
8.2	Branch-and-bound with clustering.	103
8.3	Force-displacement curves for a subset of the measured materials showing their non-linear behavior.	106
8.4	Pictures of base materials under 15 Newton force.	107
8.5	Side-by-side comparison of real and simulated materials.	109
8.6	Validation of composite materials.	110
8.7	Objet Connex 500 3D multi-material printer.	112
8.8	Validation of goal-based design algorithm.	113
8.9	Fabricated example of goal-based design.	114
8.10	Replicating a leather stool.	116
8.11	Replicating a flip flop and a slipper.	117

List of Tables

5.1	Performance numbers of our real-time animation technique for highly-detailed facial expressions.	61
7.1	Comparison of relative path-dependent errors in the work. . .	92
8.1	Error evaluation by comparing surface displacement errors of simulated and real materials.	111

Bibliography

- [Allen et al., 2002] Allen, B., Curless, B., and Popović, Z. (2002). Articulated body deformation from range scan data. *ACM Trans. Graph.*, 21(3):612–619.
- [Bando et al., 2002] Bando, Y., Kuratate, T., and Nishita, T. (2002). A simple method for modeling wrinkles on human skin. In *Proc. of Pacific Conference on Computer Graphics and Applications*, Washington, DC, USA.
- [Bathe, 1995] Bathe, K. J. (1995). *Finite Element Procedures*. Prentice Hall.
- [Becker and Teschner, 2007] Becker, M. and Teschner, M. (2007). Robust and efficient estimation of elasticity parameters using the linear finite element method. In *SimVis*, pages 15–28.
- [Beeler et al., 2010] Beeler, T., Bickel, B., Sumner, R., Beardsley, P., and Gross, M. (2010). High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (Proc. of ACM SIGGRAPH)*, 29(3).
- [Bendsoe and Sigmund, 2003] Bendsoe, M. P. and Sigmund, O. (2003). *Topology Optimization*. Springer Berlin.
- [Bergeron and Lachapelle, 1985] Bergeron, P. and Lachapelle, P. (1985). Controlling facial expression and body movements in the computer generated short “Tony de Peltrie”. *Siggraph Course Notes*.

Bibliography

- [Bickel et al., 2007] Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M. A., Pfister, H., and Gross, M. (2007). Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):33.1–33.10.
- [Bickel et al., 2008] Bickel, B., Lang, M., Botsch, M., Otaduy, M. A., and Gross, M. (2008). Pose-space animation and transfer of facial details. In *Proc. of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 57–66.
- [Blanz et al., 2003] Blanz, V., Basso, C., Poggio, T., and Vetter, T. (2003). Re-animating faces in images and video. *Computer Graphics Forum*, 22(3):641–650.
- [Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series*, pages 187–194.
- [Borshukov and Lewis, 2003] Borshukov, G. and Lewis, J. (2003). Realistic human face rendering for "The Matrix Reloaded". In *ACM SIGGRAPH 03 Sketches & Applications*.
- [Borshukov et al., 2003] Borshukov, G., Piponi, D., Larsen, O., Lewis, J., and Tempelaar-Lietz, C. (2003). Universal capture – Image-based facial animation for "The Matrix Reloaded". In *ACM SIGGRAPH 03 Sketches & Applications*.
- [Botsch et al., 2005] Botsch, M., Bommers, D., and Kobbelt, L. (2005). Efficient linear system solvers for geometry processing. In *11th IMA conference on the Mathematics of Surfaces*, pages 62–83.
- [Botsch and Kobbelt, 2004] Botsch, M. and Kobbelt, L. (2004). An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, 23(3):630–634.
- [Botsch and Kobbelt, 2005] Botsch, M. and Kobbelt, L. (2005). Real-time shape editing using radial basis functions. *Computer Graphics Forum*, 24(3):611–621.
- [Botsch and Sorkine, 2007] Botsch, M. and Sorkine, O. (2007). On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, to appear.
- [Bradley et al., 2010] Bradley, D., Heidrich, W., Popa, T., and Sheffer, A. (2010). High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (Proc. of ACM SIGGRAPH)*, 29(3).

- [Bridson et al., 2003] Bridson, R., Marino, S., and Fedkiw, R. (2003). Simulation of clothing with folds and wrinkles. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 28–36.
- [Buehler et al., 2001] Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M. (2001). Unstructured lumigraph rendering. In *Computer Graphics, SIGGRAPH 2001 Proceedings*, pages 425–432, Los Angeles, CA.
- [Burion et al., 2008] Burion, S., Conti, F., Petrovskaya, A., Baur, C., and Khatib, O. (2008). Identifying physical properties of deformable objects by using particle filters. In *Proc. of the International Conference on Robotics and Automation*, pages 1112–1117.
- [Capell et al., 2005] Capell, S., Burkhart, M., Curless, B., Duchamp, T., and Popović, Z. (2005). Physically based rigging for deformable characters. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 301–310.
- [Carr et al., 2001] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3D objects with radial basis functions. In *Proc. of ACM SIGGRAPH 01*, pages 67–76.
- [Celniker and Gossard, 1991] Celniker, G. and Gossard, D. (1991). Deformable curve and surface finite-elements for free-form shape design. In *Proc. of ACM SIGGRAPH 91*, pages 257–266.
- [Chai et al., 2003] Chai, J.-x., Xiao, J., and Hodgins, J. (2003). Vision-based control of 3D facial animation. In *Eurographics/SIGGRAPH Symposium on Computer Animation (SCA)*, pages 193–206.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.
- [DeCarlo and Metaxas, 1996] DeCarlo, D. and Metaxas, D. (1996). The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238.
- [DeCarlo and Metaxas, 2000] DeCarlo, D. and Metaxas, D. (2000). Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127.
- [d’Eon et al., 2007] d’Eon, E., Luebke, D., and Enderton, E. (2007). Efficient rendering of human skin. In *Eurographics Symposium on Rendering*, pages 147–157.

Bibliography

- [Desbrun et al., 1999] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. of ACM SIGGRAPH 99*, pages 317–324.
- [DiLorenzo et al., 2008] DiLorenzo, P. C., Zordan, V. B., and Sanders, B. L. (2008). Laughing out loud: Control for modeling anatomically inspired laughter using audio. *ACM Trans. Graph.*, 27(5):125:1–125:8.
- [Dong et al., 2010] Dong, Y., Wang, J., Pellacini, F., Tong, X., and Guo, B. (2010). Fabricating spatially-varying subsurface scattering. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(3).
- [Duchon, 1977] Duchon, J. (1977). Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In Schempp, W. and Zeller, K., editors, *Constructive Theory of Functions of Several Variables*, number 571 in Lecture Notes in Mathematics, pages 85–100. Springer.
- [Ekman and Friesen, 1978] Ekman, P. and Friesen, W. (1978). The facial action coding system: A technique for the measurement of facial movement. In *Consulting Psychologists*.
- [Essa et al., 1996] Essa, I., Basu, S., Darrell, T., and Pentland, A. (1996). Modeling, tracking and interactive animation of faces and heads: Using input from video. In *Proc. of Computer Animation 96*, pages 68–79.
- [Essa and Pentland, 1997] Essa, I. A. and Pentland, A. (1997). Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):757–763.
- [Ezzat and Poggio, 2000] Ezzat, T. and Poggio, T. (2000). Visual speech synthesis by morphing visemes. *International Journal of Computer Vision*, 38(1):45–57.
- [Fordham, 2003] Fordham, J. (2003). Middle earth strikes back. *Cinefex*, 92:71–142.
- [Galoppo et al., 2009] Galoppo, N., Otaduy, M. A., Moss, W., Sewall, J., Curtis, S., and Lin, M. C. (2009). Controlling deformable material with dynamic morph targets. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.
- [Golovinskiy et al., 2006] Golovinskiy, A., Matusik, W., Pfister, H., Rusinkiewicz, S., and Funkhouser, T. (2006). A statistical model for synthesis of detailed facial geometry. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3):1025–1034.

- [Grinspun et al., 2003] Grinspun, E., Hirani, A. N., Desbrun, M., and Schröder, P. (2003). Discrete shells. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 62–67.
- [Guenter et al., 1998] Guenter, B., Grimm, C., Wood, D., Malvar, H., and Pighin, F. (1998). Making faces. In *Proc. of ACM SIGGRAPH 98*, pages 55–66.
- [Hannon and Tiernan, 2008] Hannon, A. and Tiernan, P. (2008). A review of planar biaxial tensile test systems for sheet metal. *Journal of Materials Processing Technology*, 198(1-3):1 – 13.
- [Hart, 1967] Hart, E. W. (1967). Theory of the tensile test. *Acta Metallurgica*, 15:351–355.
- [Hasan et al., 2010] Hasan, M., Fuchs, M., Matusik, W., Pfister, H., and Rusinkiewicz, S. (2010). Physical reproduction of materials with specified subsurface scattering. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(3).
- [Hiller and Lipson, 2009] Hiller, J. and Lipson, H. (2009). Design and analysis of digital materials for physical 3d voxel printing. *Rapid Prototyping Journal*, 15:137–149.
- [Hollenstein, 2008] Hollenstein, M. (2008). Mechanical characterization of soft materials: Comparison between different experiments on synthetic specimens. Technical report, Institute of Mechanical Systems, Department of Mechanical and Process Engineering, ETH Zurich.
- [Horn, 1987] Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642.
- [Hughes, 2000] Hughes, T. J. R. (2000). *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. Dover Publications.
- [Hyneman et al., 2005] Hyneman, W., Itokazu, H., Williams, L., and Zhao, X. (2005). Human face project. In *ACM SIGGRAPH 05 Course Notes*.
- [Igarashi et al., 2005] Igarashi, T., Nishino, K., and Nayar, S. (2005). The appearance of human skin. Technical Report CUCS-024-05, Department of Computer Science, Columbia University.
- [Igarashi and Igarashi, 2009] Igarashi, Y. and Igarashi, T. (2009). Designing plush toys with a computer. *Commun. ACM*, 52(12):81–88.

Bibliography

- [Jabareen and Rubin, 2007] Jabareen, M. and Rubin, M. (2007). Hyperelasticity and physical shear buckling of a block predicted by the cosserat point element compared with inelasticity and hourglassing predicted by other element formulations. *Computational Mechanics*, 40:447–459. 10.1007/s00466-006-0119-9.
- [James and Pai, 1999] James, D. L. and Pai, D. K. (1999). Artdefo - accurate real time deformable objects. In *Proc. of SIGGRAPH 99, Computer Graphics Proceedings*, pages 65–72.
- [Joly, 2010] Joly, J. (2010). Can a polygon make you cry. World Wide Web electronic publication, <http://www.jonathanjoly.com/front.htm>. [Online; accessed 01-July-2010].
- [Jones et al., 2006] Jones, A., Gardner, A., Bolas, M., McDowall, I., and Debevec, P. (2006). Performance geometry capture for spatially varying re-lighting. In *3rd European Conference on Visual Media Production (CVMP 2006)*.
- [Joshi et al., 2003] Joshi, P., Tien, W. C., Desbrun, M., and Pighin, F. (2003). Learning controls for blend shape based realistic facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 187–192. Eurographics Association.
- [Kajberg and Lindkvist, 2004] Kajberg, J. and Lindkvist, G. (2004). Characterisation of materials subjected to large strains by inverse modelling based on in-plane displacement fields. *International Journal of Solids and Structures*, 41(13):3439–3459.
- [Kauer et al., 2002] Kauer, M., Vuskovic, V., Dual, J., Szekely, G., and Bajka, M. (2002). Inverse finite element characterization of soft tissues. *Medical Image Analysis*, 6(3):257–287.
- [Kaufmann et al., 2008] Kaufmann, P., Martin, S., Botsch, M., and Gross, M. (2008). Flexible simulation of deformable models using discontinuous galerkin fem. *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 105–115.
- [Kharevych et al., 2009] Kharevych, L., Mullen, P., Owhadi, H., and Desbrun, M. (2009). Numerical coarsening of inhomogeneous elastic materials. *ACM Trans. Graph.*, 28(3):51:1–51:8.
- [Kähler et al., 2001] Kähler, K., Haber, J., and Seidel, H. (2001). Geometry-based muscle modeling for facial animation. In *In Proc. Graphics Interface 2001*, pages 37–46.

- [Kicinger et al., 2005] Kicinger, R., Arciszewski, T., and Jong, K. D. (2005). Evolutionary computation and structural design: A survey of the state-of-the-art. *Comput. Struct.*, 83(23-24):1943–1978.
- [Kim and Vendrovsky, 2008] Kim, T.-Y. and Vendrovsky, E. (2008). Driven-shape - a data-driven approach to shape deformation. In *Proc. of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*.
- [Klaudiny et al., 2010] Klaudiny, M., Hilton, A., and Edge, J. (2010). High-detail 3d capture of facial performance. In *3DPVT*.
- [Kobbelt et al., 1999] Kobbelt, L., Vorsatz, J., and Seidel, H.-P. (1999). Multiresolution hierarchies on unstructured triangle meshes. *Comput. Geom. Theory Appl.*, 14(1-3):5–24.
- [Koch et al., 1996] Koch, R. M., Gross, M. H., Carls, F. R., von Büren, D. F., Fankhauser, G., and Parish, Y. (1996). Simulating facial surgery using finite element methods. In *Proc. of ACM SIGGRAPH 96*, pages 421–428.
- [Kry et al., 2002] Kry, P., James, D. L., and Pai, D. K. (2002). EigenSkin: Real-time large deformation character skinning in hardware. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 153–159.
- [Kry and Pai, 2006] Kry, P. G. and Pai, D. K. (2006). Interaction capture and synthesis. *ACM Trans. Graph. (Proc. of ACM SIGGRAPH)*, 25(3):872–880.
- [Kurihara and Miyata, 2004] Kurihara, T. and Miyata, N. (2004). Modeling deformable human hands from medical images. In *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 357–366.
- [Land and Doig, 1960] Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520.
- [Lang et al., 2002] Lang, J., Pai, D. K., and Woodham, R. J. (2002). Acquisition of elastic models for interactive simulation. *International Journal of Robotics Research*, 21(8):713–733.
- [Lanir, 1987] Lanir, Y. (1987). Skin mechanics. In Skalak, R. and Chien, S., editors, *Handbook of Bioengineering*, pages 11.1–11.25. McGraw-Hill.
- [Lee and Terzopoulos, 2006] Lee, S.-H. and Terzopoulos, D. (2006). Heads up!: biomechanical modeling and neuromuscular control of the neck. *ACM Trans. Graph.*, 25(3):1188–1198.

Bibliography

- [Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168.
- [Levoy et al., 2000] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. (2000). The digital michelangelo project: 3D scanning of large statues. In *Computer Graphics, SIGGRAPH 2000 Proceedings*, pages 131–144, Los Angeles, CA.
- [Lewis et al., 2000] Lewis, J. P., Cordner, M., and Fong, N. (2000). Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of ACM SIGGRAPH*, pages 165–172.
- [Li et al., 1993] Li, H., Roivainen, P., and Forchheimer, R. (1993). 3-D motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(6):545–555.
- [Li et al., 2010a] Li, H., Weise, T., and Pauly, M. (2010a). Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010)*, 29(3).
- [Li et al., 2010b] Li, X.-Y., Shen, C.-H., Huang, S.-S., Ju, T., and Hu, S.-M. (2010b). Popup: automatic paper architectures from 3d models. *ACM Trans. Graph.*, 29(4):1–9.
- [Liou, 2007] Liou, F. W. (2007). *Rapid prototyping and engineering applications: a toolbox for prototype development*. CRC Press.
- [Lund and Stegmann, 2005] Lund, E. and Stegmann, J. (2005). On structural optimization of composite shell structures using a discrete constitutive parameterization. *Wind Energy*, 8:109–124.
- [Ma et al., 2007] Ma, W.-C., Hawkins, T., Peers, P., Chabert, C.-F., Weiss, M., and Debevec, P. (2007). Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proc. Eurographics Symposium on Rendering*, pages 183–194.
- [Ma et al., 2008] Ma, W.-C., Jones, A., Chiang, J.-Y., Hawkins, T., Frederiksen, S., Peers, P., Vukovic, M., Ouhyoung, M., and Debevec, P. (2008). Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph. (Proc. of ACM SIGGRAPH Asia)*, 27(5).
- [Magnenat-Thalmann et al., 2002] Magnenat-Thalmann, N., Kalra, P., Lévêque, J. L., Bazin, R., Batische, D., and Queleux, B. (2002). A computational skin model: fold and wrinkle formation. *IEEE Trans. on Information Technology in Biomedicine*, 6(4):317–323.

- [Magenat-Thalmann et al., 1988] Magenat-Thalmann, N., Laperriere, R., and Thalmann, D. (1988). Joint-dependent local deformations for hand animation and object grasping. In *Proc. of Graphics Interface*, pages 317–323.
- [Matusik et al., 2009] Matusik, W., Ajdin, B., Gu, J., Lawrence, J., Lensch, H. P., Pellacini, F., and Rusinkiewicz, S. (2009). Printing spatially-varying reflectance. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 28(5).
- [Matusik et al., 2003] Matusik, W., Pfister, H., Brand, M., and McMillan, L. (2003). A data-driven reflectance model. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):759–770.
- [Meyer and Anderson, 2007] Meyer, M. and Anderson, J. (2007). Key point subspace acceleration and soft caching. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):74.
- [Meyer et al., 2003] Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In Hege, H.-C. and Polthier, K., editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg.
- [Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.
- [Mitani and Suzuki, 2004] Mitani, J. and Suzuki, H. (2004). Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.*, 23(3):259–263.
- [Mori, 1970] Mori, M. (1970). Bukimi no tani - the uncanny valley. *Energy*, 7:33–35.
- [Mori and Igarashi, 2007] Mori, Y. and Igarashi, T. (2007). Plushie: An interactive design system for plush toys. *ACM Trans. Graph.*, 26(3):45:1–45:8.
- [Müller and Gross, 2004] Müller, M. and Gross, M. H. (2004). Interactive virtual materials. In *Graphics Interface 2004*, pages 239–246.
- [Nava, 2007] Nava, A. (2007). *In vivo characterization of the mechanical response of soft human tissue*. PhD thesis, ETH Zurich, Zurich. Dissertation Number 17060.
- [Nava et al., 2003] Nava, A., Mazza, E., Kleinermann, F., Avis, N. J., and McClure, J. (2003). Determination of the mechanical properties of soft human tissues through aspiration experiments. In *Proc. of MICCAI*, pages 222–229.

Bibliography

- [Nealen et al., 2006] Nealen, A., Müller, M., Keiser, R., Boxermann, E., and Carlson, M. (2006). Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836.
- [Nehab et al., 2005] Nehab, D., Rusinkiewicz, S., Davis, J., and Ramamoorthi, R. (2005). Efficiently combining positions and normals for precise 3d geometry. *ACM Transactions on Graphics*, 24(3):536–543.
- [Nesme et al., 2009] Nesme, M., Kry, P. G., Jerábková, L., and Faure, F. (2009). Preserving topology and elasticity for embedded deformable models. *ACM Trans. Graph.*, 28(3):52:1–52:9.
- [Neumaier and Pownuk, 2007] Neumaier, A. and Pownuk, A. (2007). Linear systems with large uncertainties with applications to truss structures. *Reliable Computing*, 13:149–172.
- [Noh and Neumann, 1999] Noh, J.-Y. and Neumann, U. (1999). A survey of facial modeling and animation techniques. Technical Report USC-TR-99-705, University of Southern California.
- [Noh and Neumann, 2001] Noh, J.-y. and Neumann, U. (2001). Expression cloning. In *Proc. of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 277–288.
- [NVIDIA, 2008] NVIDIA (2008). <http://developer.nvidia.com/cuda>.
- [OBJET, 2009] OBJET (2009). Connex500 multi-material 3d printing system. <http://www.objet.com/3D-Printer/Connex500/>.
- [Ogden, 1997] Ogden, R. W. (1997). *Non-Linear Elastic Deformations*. Courier Dover Publications.
- [Okabe et al., 1992] Okabe, H., Imaoka, H., Tomiha, T., and Niwaya, H. (1992). Three dimensional apparel cad system. In *Computer Graphics (Proc. of SIGGRAPH 92)*, pages 105–110.
- [Ostermann, 1998] Ostermann, J. (1998). Animation of synthetic faces in mpeg-4. In *CA '98: Proceedings of the Computer Animation*, page 49, Washington, DC, USA. IEEE Computer Society.
- [Ottensmeyer and Salisbury Jr., 2004] Ottensmeyer, M. P. and Salisbury Jr., J. K. (2004). In-vivo data acquisition instrument for solid organ mechanical property measurement. In *Proc. of MICCAI*, pages 975–982.
- [Pai et al., 2001] Pai, D. K., van den Doel, K., James, D. L., Lang, J., Lloyd, J. E., Richmond, J. L., and Yau, S. H. (2001). Scanning physical interaction behavior of 3d objects. In *Proceedings of ACM SIGGRAPH*, pages 87–96.

- [Park and Hodgins, 2006] Park, S. I. and Hodgins, J. K. (2006). Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*, 25(3).
- [Park and Hodgins, 2008] Park, S. I. and Hodgins, J. K. (2008). Data-driven modeling of skin and muscle deformation. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*, 27(3).
- [Parke, 1974] Parke, F. I. (1974). *A parametric model for human faces*. PhD thesis, University of Utah, Salt Lake City, Utah.
- [Pighin et al., 1998] Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., and Salesin, D. (1998). Synthesizing realistic facial expressions from photographs. In *Computer Graphics*, volume 32 of *SIGGRAPH 98 Proceedings*, pages 75–84.
- [Pighin et al., 1999] Pighin, F. H., Szeliski, R., and Salesin, D. (1999). Resynthesizing facial animation through 3D model-based tracking. In *International Conference on Computer Vision (ICCV)*, pages 143–150.
- [Rebonato and Jäckel, 1999] Rebonato, R. and Jäckel, P. (1999). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. Technical report, Quantitative Research Centre, NatWest Group.
- [Rhee et al., 2006] Rhee, T., Lewis, J. P., and Neumann, U. (2006). Real-time weighted pose-space deformation on the GPU. *Computer Graphics Forum (Proc. Eurographics)*, 25(3):439–448.
- [Schnur and Zabarar, 1992] Schnur, D. S. and Zabarar, N. (1992). An inverse method for determining elastic material properties and a material interface. *International Journal for Numerical Methods in Engineering*, 33(10):2039–2057.
- [Schoner et al., 2004] Schoner, J. L., Lang, J., and Seidel, H.-P. (2004). Measurement-based interactive simulation of viscoelastic solids. *Computer Graphics Forum*, 23(3):547–556.
- [Shi et al., 2006] Shi, L., Yu, Y., Bell, N., and Feng, W.-W. (2006). A fast multigrid algorithm for mesh deformation. *ACM Transactions on Graphics*, 25(3):1108–1117.
- [Sifakis et al., 2005] Sifakis, E., Neverov, I., and Fedkiw, R. (2005). Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics*, 24(3):417–425.

Bibliography

- [Sloan et al., 2001] Sloan, P.-P. J., Rose, III, C. F., and Cohen, M. F. (2001). Shape by example. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 135–143, New York, NY, USA. ACM.
- [Sueda et al., 2008] Sueda, S., Kaufman, A., and Pai, D. K. (2008). Musculo-tendon simulation for hand animation. *ACM Trans. Graph.*, 27(3):83:1–83:8.
- [Sumner and Popović, 2004] Sumner, R. W. and Popović, J. (2004). Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405.
- [Sumner et al., 2005] Sumner, R. W., Zwicker, M., Gotsman, C., and Popović, J. (2005). Mesh-based inverse kinematics. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 24(3):488–495.
- [Svoboda et al., 2005] Svoboda, T., Martinec, D., and Pajdla, T. (2005). A convenient multicamera self-calibration for virtual environments. *Presence: Teleoper. Virtual Environ.*, 14(4):407–422.
- [Teran et al., 2005] Teran, J., Sifakis, E., Blemker, S. S., Ng-Thow-Hing, V., Lau, C., and Fedkiw, R. (2005). Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans. on Visualization and Computer Graphics*, 11(3):317–328.
- [Terzopoulos et al., 1987] Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically deformable models. In *Computer Graphics (Proc. of SIGGRAPH 87)*, pages 205–214.
- [Terzopoulos and Waters, 1993] Terzopoulos, D. and Waters, K. (1993). Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14:569–579.
- [Toledo et al.,] Toledo, S., Chen, D., and Rotkin, V. Taucs: A library of sparse linear solvers. <http://www.tau.ac.il/~stoledo/taucs>.
- [Toledo et al., 2003] Toledo, S., Chen, D., and Rotkin, V. (2003). Taucs: A library for sparse linear solvers.
- [Venkataraman et al., 2005] Venkataraman, K., Lodha, S., and Raghavan, R. (2005). A kinematic-variational model for animating skin with wrinkles. *Computers & Graphics*, 29(5):756–770.
- [Vlasic et al., 2005] Vlasic, D., Brand, M., Pfister, H., and Popović, J. (2005). Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433.

- [Wang et al., 2007] Wang, R. Y., Pulli, K., and Popović, J. (2007). Real-time enveloping with rotational regression. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):73.
- [Wang et al., 2004] Wang, Y., Huang, X., Lee, C.-S., Zhang, S., Li, Z., Samaras, D., Metaxas, D., Elgammal, A., and Huang, P. (2004). High resolution acquisition, learning and transfer of dynamic 3-D facial expressions. *Computer Graphics Forum*, 23(3):677–686.
- [Weber et al., 2007] Weber, O., Sorkine, O., Lipman, Y., and Gotsman, C. (2007). Context-aware skeletal shape deformation. *Computer Graphics Forum (Proc. Eurographics)*, 26(3):265–274.
- [Weise et al., 2009] Weise, T., Li, H., Gool, L. V., and Pauly, M. (2009). Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*. Eurographics Association.
- [Wenger et al., 2005] Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., and Debevec, P. (2005). Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics*, 24(3):756–764.
- [Weyrich et al., 2006] Weyrich, T., Matusik, W., Pfister, H., Bickel, B., Donner, C., Tu, C., McAndless, J., Lee, J., Ngan, A., Jensen, H. W., and Gross, M. (2006). Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics*, 25(3):1013–1024.
- [Weyrich et al., 2009] Weyrich, T., Peers, P., Matusik, W., and Rusinkiewicz, S. (2009). Fabricating microgeometry for custom surface reflectance. *ACM Trans. Graph.*, 28(3).
- [Williams, 1990] Williams, L. (1990). Performance-driven facial animation. In *Computer Graphics (Proceedings of SIGGRAPH 90)*, pages 235–242.
- [Wu et al., 1996] Wu, Y., Kalra, P., and Magnenat-Thalmann, N. (1996). Simulation of static and dynamic wrinkles of skin. In *Proc. of Computer Animation*, pages 90–97.
- [Zhang, 2004] Zhang, H. (2004). Discrete combinatorial laplacian operators for digital geometry processing. In *Proc. of SIAM Conference on Geometric Design and Computing*, pages 575–592. Nashboro Press.
- [Zhang et al., 2004] Zhang, L., Snavely, N., Curless, B., and Seitz, S. M. (2004). Spacetime faces: High resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558.

Bibliography

- [Zhang and Huang, 2006] Zhang, S. and Huang, P. (2006). High-resolution, real-time three-dimensional shape measurement. *Optical Engineering*, 45(12).
- [Zhang and Sim, 2005] Zhang, Y. and Sim, T. (2005). Realistic and efficient wrinkle simulation using an anatomy-based face model with adaptive refinement. In *Computer Graphics International 2005*, pages 3–10.
- [Zohdi and Wriggers, 2004] Zohdi, T. I. and Wriggers, P. (2004). *Introduction to Computational Micromechanics*. Springer-Verlag New York, Inc.
- [Zordan et al., 2004] Zordan, V. B., Celly, B., Chiu, B., and DiLorenzo, P. C. (2004). Breathe easy: model and control of simulated respiration for animation. In *2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 29–37.