FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

RNDr. Petr Kadleček

Inverse Anatomical Modeling and Simulation of Virtual Humans

Department of Software and Computer Science Education

Supervisor of the doctoral thesis:	Mgr. Ladislav Kavan, Ph.D.
Study programme:	Informatics
Study branch:	Visual computing and computer games

Prague 2022

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to express my sincere gratitude to my advisor Ladislav Kavan. He invited me to join his lab at the University of Utah and dedicated a lot of his time to teach me all about physics-based animation and research in general. I especially enjoyed our research discussions when hiking in Rocky Mountains.

I would not even start my PhD without Jaroslav Křivánek, who unfortunately left the world too early. He brought a lot of enthusiasm for computer graphics to our group in Prague and changed my career and life in just one short conversation. His presence is dearly missed.

I am deeply grateful for the extraordinary collaboration with Alexandru Eugen Ichim and his advisor Mark Pauly from EPFL in Lausanne. Alex's determination allowed us to focus on important problems and eventually publish our work despite being more than 5000 miles away from each other.

Many thanks to Tiantian Liu for helping me understand fundamentals of the finite element method and for all the research discussions and fun we had together with other lab mates in Utah including Dimitar Dinev, Jing Li, Riddhish Bhalodia and Junior Rojas to name a few. I also thank for all the feedback and discussions in the Utah Graphics Seminar with Cem Yuksel, Erik Brunvand and other faculty members, students and alumni in Utah.

I would like to thank for all the support from the staff at the The J. Willard Marriott Library and the Department of Radiology and Imaging Sciences at the University of Utah. Without their help, I would not be able to collect 3D surface scan and MRI data necessary for my research projects. I would also like to thank Angel Torrado Carvajal from the Harvard Medical School who kindly processed some of our research data using the Pseudo-CT method.

My appreciation goes to Josef Pelikán, Alexander Wilkie and the entire Computer Graphics Group of the Charles University in Prague. Their great lectures and passion for computer graphics were a great motivation to pursue this direction in my life. I would like to thank my bachelor and master thesis supervisor Petr Kmoch. I would not find my way to wonders of physics-based simulation without his guidance. Further, I thank my brother-in-law Martin Kahoun for inspirational discussions, proof-reading my grant proposals, papers and theses.

My deepest gratitude goes to my family: my parents Boleslav and Wanda for their incredible care and support, my brothers Vojta and Tomáš for introducing me to programming and graphics.

Finally, a very special thanks goes to my wife Tereza, who helped me follow my dreams around the world and endured challenging times far from friends and family. I would have not finished this thesis without her endless support and love.

The research in this thesis was supported by The Charles University Grant Agency (projects GA UK 2062214 and GA UK 1524217).

Title: Inverse Anatomical Modeling and Simulation of Virtual Humans

Author: RNDr. Petr Kadleček

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Ladislav Kavan, Ph.D., Department of Software and Computer Science Education

Abstract:

In this dissertation, we focus on a mechanical aspect of a human body and face modeling. We leverage existing physics-based models of elasticity and use them as building blocks to create an animatable virtual human. We model different types of hard and soft tissues to enable our model to contract muscles, interact with an environment or realistically deform when subjected to external forces. In the first part of the work, we present a method to create personalized anatomical models of human body ready for physics-based animation, using only a set of 3D surface scans. We start by building a template anatomical model of an average male which supports deformations due to both 1) subject-specific variations: shapes and sizes of bones, muscles, and adipose tissues and 2) skeletal poses. Next, we capture a set of 3D scans of an actor in various poses. Our key contribution is formulating and solving a large-scale optimization problem where we compute both subject-specific and pose-dependent parameters such that our resulting anatomical model explains the captured 3D scans as closely as possible. Compared to data-driven body modeling techniques that focus only on the surface, our approach has the advantage of creating physics-based models, which provide realistic 3D geometry of the bones and muscles, and naturally supports effects such as inertia, gravity, and collisions according to Newtonian dynamics. The second part of the thesis focuses on the inverse facial modeling. The human face is an anatomical system exhibiting heterogenous and anisotropic mechanical behavior. This leads to complex deformations even in a neutral facial expression due to external forces such as gravity. To obtain data on facial deformations we capture and register 3D scans of the face with different gravity directions and with various facial expressions. We show two approaches of model building either from an anatomical template or leveraging data from magnetic resonance imaging for more accurate modeling. Our main contribution consists in formulating and solving an inverse physics problem where we learn mechanical properties of the face and match expressions by novel muscle activation models while taking into account collisions. We demonstrate that our model generates predictions of facial deformations more accurately than recent related physics-based techniques.

Keywords: computer animation, physics-based simulation, anatomical modeling, numerical optimization

Contents

1. Introduction

Computational models play a key role in many aspects of our lives. We utilize them to understand and predict behavior of complex systems and they enable us to experiment with scenarios that might not be feasible in the real world. One such vastly complex system is the human body. To create an ultimate virtual human, we would ideally model all anatomical details, functioning of each human cell and dynamics of all its molecules as a single system. Unfortunately, this is still beyond our processing capabilities despite the exponential growth of computational power in last decades.

In this work, we focus on a mechanical aspect of a human body modeling. We leverage existing physics-based models of elasticity and use them as building blocks to create an animatable virtual human. We model different types of hard and soft tissues to enable our model to contract muscles, interact with an environment or realistically deform when subjected to external forces.

There are many interesting applications of virtual human models. They are especially useful in scenarios that are not practical for real world experiments or that would require laborious prototyping, e.g., a virtual surgery training, crash tests for a safe vehicle design, ergonomy evaluation, injury analysis or prosthetic design. More recently, virtual humans are used as digital doubles in a movie industry for visual effects, computer games and in a virtual or augmented reality to reconstruct a realistic appearance of a person in a virtual environment.

The aim of this dissertation is to explore methods for an automated personalized virtual human reconstruction of both human body and face models mainly intended for applications in computer graphics and animation. The output of these methods consists of animatable digital characters represented as volumetric meshes augmented with anatomically-based tissue and physics-based material properties ready for a simulation.

1.1 Modeling a Virtual Human

One notable difference of models presented in this thesis to some existing biomechanical models is that we are not concerned about accuracy of predicted forces to the degree required for, e.g., analytic applications. The important factor is, rather, a visual plausibility of model's behavior. In our case, this represents believability of deformations and dynamics on a surface of a body or face. This allows us to simplify some aspects of the modeling and simulate complexities of large scale (i.e. complete body and face) models as shown in this work. Nevertheless, creating believable body and face model of a person is still a daunting task. Let us describe an example process of creating a digital double by technical artists.

The process starts with a single high quality 3D surface scan of a full body represented by a 3D point cloud (e.g., captured using a photogrammetric setup of multiple digital cameras [Mikhail et al., 2001]). The point cloud is manually cleaned and turned into a polygonal surface mesh at a desired resolution onto which a texture is mapped [Fabio et al., 2003]. An artist proceeds with a fitting of a skeletal rig and mapping skinning weights of each joint that affect deformation of the body surface due to body pose changes [Kavan et al., 2007]. At this point, a skilled artist manually estimates the best location of each body joint to minimize deformation artifacts. Additional surface details can be added by auxiliary mesh warping tools, sometimes referred to as deformers. An example of a deformer is a bulging of a muscle during its contraction. Artists can employ muscle modeling systems that can help with creating such deformers. These systems, however, still require a lot of manual work and skill to produce faithful results. Automating this process for each individual is a significant challenge and the large amount of work is needed for every single personalized model. Technical artists often spend thousands of hours on a digital double appearing in a film just for a few seconds [Dunlop, 2014].

Special attention is given to the modeling of a face. Humans are particularly sensitive to facial expressions since they act as a mean of communication. For this reason, the process of a facial model reconstruction requires more input data. A capture session typically includes tens or even hundreds of different expressions. An example set of expressions can be found in a system of facial expression classification FACS (facial action coding system), which was inspired by facial muscle group movements [Ekman and Rosenberg, 1997]. This set of expressions is then used to create complex blendshapes that together with a manually rigged temporomandibular joint drive the deformation of a face model. More advanced effects such as muscle contractions, dynamics or deformation due to external forces is rarely modeled due to the additional difficulty [Cong et al., 2015, 2016].

1.2 Physics-based inverse problems

Motivated by these difficulties, we attempt to address the issue of creating personalized physics-based virtual humans by formulating the modeling task as an optimization problem given a sparse set of observations. We employ models of elasticity and anatomical priors to help us control many ambiguities arising from using only sparse input data.

Our approach to animation of virtual humans is based on a combination of standard computer animation techniques and physics-based anatomical models. Specifically, the skeleton of a virtual human is controlled by the user or existing animation data but the soft tissue connected to the skeleton and its deformation is fully simulated. We refer to this method as the physics-based skinning or the forward problem. An input of the forward problem consists of a personalized virtual human model and animation data (e.g., joint angles and muscle activations). The output of the simulation is an animated virtual human which can be represented as a mesh per animation frame.

The inverse problem, which is the core of this work, attempts to find a personalized virtual human model given a sparse set of surface meshes of a person in different poses or making different facial expressions. The method works by adapting a general anatomical template defining a structure of the skeleton, muscles and fat such that the adapted template best explains all input frames, e.g., surface scans of a person in different poses. The adaptation is inspired by anatomical models of growth and existing statistical data and is formulated as a large constrained optimization problem described in detail in this thesis.

2. Simulation of soft tissues

2.1 Introduction

The main concept of our approach to simulating soft tissues is based on continuum mechanics. In this branch of physics, objects of different materials are modeled as a continuous mass rather than nanoscale particles that form complex bindings. This approximation of a material might be too crude for some applications, e.g. in molecular dynamics. In our case, modeling every molecule in a human body would not only be impractical from the perspective of a mechanical modeling but also computationally infeasible when simulating large parts of a body.

In continuum mechanics, a deformation of soft tissues due to applied forces can be formulated as a relation between variables representing a change of positions of infinitesimal material points within a tissue continuum (strain) and a physical quantity representing internal tissue forces (stress). The stress-strain relation, also called a constitutive equation, typically forms a system of partial differential equations. These equations are usually too complex to be solved analytically and various numerical approaches are used. The most popular method, which we also rely on, is the finite element method (FEM) that involves spatial discretization of a continuum into finite elements.

The rest of this chapter covers a brief background of a FEM-based simulation and introduces specific materials and naming conventions used in this dissertation. Please refer to practical course notes [Sifakis and Barbic, 2012] for a more comprehensive overview or [Bonet and Wood, 1997, Müller et al., 2008, Maas et al., 2012] to learn more details.

2.2 Finite Element Method

2.2.1 Deformation of linear tetrahedral elements

Our soft tissue models are discretized into tetrahedral elements (see Figure 2.1). We generate a tetrahedral mesh from a triangular surface mesh and optional internal constraints by invoking a constrained Delaunay tetrahedralization with predefined Steiner points implemented by TetGen [Si, 2015]. Material points of the continuum are then approximated by an interpolation of nodal values within each tetrahedron. In our case, nodes correspond to vertices of a tetrahedral mesh and material point values within each element are interpolated linearly.

Mathematically, we can describe a deformation as a map from an undeformed configuration to a deformed one. For a material point $\vec{\mathbf{X}}$ in an undeformed configuration, we define a material point in a deformed state as $\vec{\mathbf{x}} = \phi(\vec{\mathbf{X}})$. In our case, the deformation map ϕ of the continuum in a tetrahedron i is defined as a following linear map:

$$\vec{\mathbf{x}} = \phi_i(\vec{\mathbf{X}}) = \mathbf{A_i}\vec{\mathbf{X}} + \vec{\mathbf{b_i}} \tag{2.1}$$

where $\mathbf{A_i} \in \mathbb{R}^{3\times3}$ is a linear transformation and $\vec{\mathbf{b_i}} \in \mathbb{R}^{3\times1}$ a translation vector specific to an element i. The full continuum deformation map that covers all

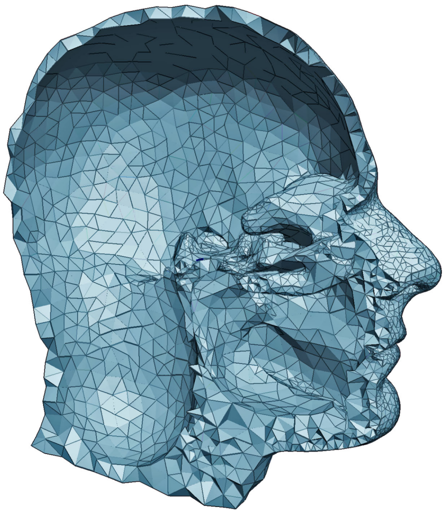


Figure 2.1: An example of a sliced tetrahedral mesh of a human head.

material points within the deformable object is a piecewise linear function over all non-overlapping tetrahedral elements.

The Jacobian of this map, which is constant in the linear case, is referred to as a deformation gradient:

$$\mathbf{F_i} = \frac{\partial \phi_i(\vec{\mathbf{X}})}{\partial \vec{\mathbf{X}}} = \mathbf{A_i} \in \mathbb{R}^{3\times3} \tag{2.2}$$

We use the deformation gradient as a quantity parameterizing constitutive models of different elastic materials in Section 2.2.2. The equation above also gives us a good intuition that a deformation gradient in this case is simply a linear transformation of a tetrahedral element in an undeformed configuration and equals identity in the case of no deformation.

For purposes of this thesis, it is convenient to express the deformation gradient as a function of both deformed and undeformed configurations of tetrahedron vertices. We first define a shape matrix function as:

$$\mathbf{D}(\mathbf{p}) = \begin{bmatrix}\vec{\mathbf{p_1}} - \vec{\mathbf{p_4}} & \vec{\mathbf{p_2}} - \vec{\mathbf{p_4}} & \vec{\mathbf{p_3}} - \vec{\mathbf{p_4}}\end{bmatrix} \in \mathbb{R}^{3\times3} \tag{2.3}$$

where $\vec{\mathbf{p_j}}$ is the jth vertex of a tetrahedron. The deformation gradient can now be expressed as

$$\mathbf{F_i} = \mathbf{D}(\mathbf{x})\mathbf{D}(\mathbf{X})^{-1} \tag{2.4}$$

where $\mathbf{x} \in \mathbb{R}^{3\times3}$ contains vertices of a tetrahedron i in a deformed configuration and $\mathbf{X} \in \mathbb{R}^{3\times3}$ in an undeformed one (see [Sifakis and Barbic, 2012] for a derivation).

2.2.2 Elastic materials

Capturing various material behavior is a long-standing open problem in physics and materials science [Beatty, 1987, Ogden et al., 2004, Maas et al., 2012]. Various types of tissues and even the same tissue material under distinct conditions exhibit different mechanical properties. These conditions, for example, include temperature or hydration of a tissue [Shahmirzadi et al., 2013]. The stress-strain relationship, which is often non-linear, can also depend on factors such as an orientation, creating a group of anisotropic materials. For example, a fibrous tissue material made of collagens, such as connective tissue which represents most of the mechanical behavior of soft tissues [Chagnon et al., 2015], exhibits different behavior depending on a direction of an applied stress [Ogden, 2003]. Another example is a muscle tissue which is often modeled as a transversely-isotropic material with one axis of symmtery depending on an orientation of muscle fibers [Blemker and Delp, 2005].

In this thesis, we define stress-strain relationships via strain energy formulations parametrized by a deformation gradient (Eq. 2.2). We are mostly focused on hyperelastic materials that can recover even from large deformations, i.e., we do not model deformations causing tissue damage and injuries. The basic principle of a strain energy formulation is to find a differentiable function that can best capture diverse modes of deformation of a class of objects while keeping a value of the function due to rigid transformation ideally zero. Since the strain energy in our case is a function of a deformation gradient, it is already invariant to translations. Rotation invariance is handled in various ways depending on the indended performance and behavior.

Many constitutive models are also parameterized by material-specific coefficients within a modeled class of objects (e.g. a soft vs. hard rubber). For example, Young's modulus E typically expressed in the physical unit of gigapascals corresponds to a material stiffness and Poisson's ratio ν with no physical unit measures how does an object deform in a direction perpendicular to the applied force. An object with a Poisson's ratio $\nu = 0.5$ is perfectly incompressible, whereas an object with $\nu = 0$ does not show the Poisson effect. Since these coefficients are often estimated from a series of experiments [Ogden et al., 2004], they can vary between different constitutive models. The estimation becomes even more challenging for soft tissues where various approaches to measure the stress-strain relationship (e.g. in-vivo and in-vitro) provide different results [Lapeer et al., 2011].

2.2.3 Linear elasticity

Linear elasticity, as the name suggests, has a linear stress-strain relationship. The main benefit of linear elasticity model is its computational performance. The complete quasi-static system can be solved in only one iteration as we show later in Section 2.3. The downside of the linearization is that large rotations are penalized as deformations. However, this might not be a significant problem in some cases, such as deformations of facial expressions, making this material practical for initial experiments and prototyping where performance can be an issue.

The definition of linear elasticity in terms of a strain energy of all tetrahedral elements is following:

$$\mathbf{E}_{Linear}(\mathbf{F}) = \sum_{i=1} w_i \mu_i \| \frac{1}{2}(\mathbf{F}_i + \mathbf{F}_i^T) - \mathbf{I}\|_F^2 + \sum_{i=1} w_i \frac{\lambda_i}{2} tr(\frac{1}{2}(\mathbf{F}_i + \mathbf{F}_i^T) - \mathbf{I})^2 \quad (2.5)$$

where $\mathbf{I} \in \mathbb{R}^{3\times3}$ is identity, $\mathbf{F}_i$ is a deformation gradient and $w_i = \frac{1}{6}|\det \mathbf{D}(\mathbf{X})| \in \mathbb{R}$ is a volume of a tetrahedral element i and $\|.\|_F$ denotes Frobenius norm. Scalar quantities μ_i and λ_i are material-dependent quantities known as Lamé coefficients that are related to Young's modulus E and Poisson's ratio ν as follows:

$$\mu = \frac{E}{2(1+\nu)} \tag{2.6}$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{2.7}$$

Note that the linear elasticity formulation is usually simplified by introducing a small strain tensor $\epsilon = \frac{1}{2}(\mathbf{F}_i + \mathbf{F}_i^T) - \mathbf{I}$. Please refer to [Sifakis and Barbic, 2012] for more details.

2.2.4 Corotational linear elasticity

One of the more widely adopted constitutive model especially in computer graphics is Corotational linear elasticity [Etzmuß et al., 2003, Müller and Gross, 2004, McAdams et al., 2011b]. The idea of this model is to factor-out a rotation $\mathbf{R} \in SO(3)$ from the deformation gradient $\mathbf{F}$ computed for each element to achieve the rotation invariance:

$$\mathbf{E}_{Corot}(\mathbf{F}) = \sum_{i=1} w_i \mu_i \|\mathbf{F}_i - \mathbf{R}_i\|_F^2 + \sum_{i=1} w_i \frac{\lambda_i}{2} tr(\mathbf{R}_i^T \mathbf{F}_i - \mathbf{I})^2 \tag{2.8}$$

We obtain the rotation matrix $\mathbf{R}$ by applying the singular value decomposition on the deformation gradient [McAdams et al., 2011a] and making sure that matrix $\mathbf{R}$ is orthonormal with positive determinant [Twigg and Kacic-Alesic, 2010]. A variant of this energy for triangular meshes using the first term of Eq. 2.8 is also referred to as ”as-rigid-as-possible” [Sorkine and Alexa, 2007] (or ARAP).

2.2.5 Rubber-like materials

Biological tissues, such as adipose tissue, exhibit a non-linear behavior under large loads that the previous constitutive models do not capture [Famaey and Sloten, 2008, Mihai et al., 2015]. In applications where such deformations are important, a more sophisticated class of polynomial models can be used. A large class of models was formulated to capture rubber-like materials as they represent highly non-linear properties that can be easily observed and measured. These models include Ogden's [Ogden, 1972, Ogden and Holzapfel, 2006] material model or a Mooney-Rivlin model [Mooney, 1940] and have various complexity depending on the number of parameters used. The Mooney-Rivlin model, for example, can be parameterized by up to nine coefficients, making the model fitting rather complicated. Both these models can be reduced to a less complicated constitutive model named Neo-Hookean [Bonet and Wood, 1997] widely used in many fields including computer animation for its flesh-like deformation properties. A variant of this material called stable Neo-Hookean model that was proposed to overcome specific numerical challenges in a flesh simulation [Smith et al., 2018] is formulated as follows:

$$\mathbf{E}_{Neo}(\mathbf{F}) = \sum_{i=1} w_i \frac{\mu_i}{2}(\mathbf{I}_C^i - 3) + \sum_{i=1} w_i \frac{\lambda_i}{2}(det(\mathbf{F}_i) - \alpha)^2 - \sum_{i=1} w_i \frac{\mu_i}{2} log(I_C^i + 1) \tag{2.9}$$

where $\mathbf{I}_C = tr(\mathbf{F}^T\mathbf{F})$, $\alpha = 1 + \mu/\lambda - (\mu/4)\lambda$ and i is the element number.

2.2.6 Muscle activation model

Variety of constitutive models for both active and passive response of muscle tissue exist [Weiss et al., 1996, Teran et al., 2003]. Most of these models assume that a fiber direction for each element and a specific muscle is known. The fiber direction can be obtained from existing artist-made models, diffusion tensor imaging (DTI) [Damon et al., 2017] using data from a magnetic resonance imaging (MRI) device, ultrasonographic measurements and a cadaver analysis [Martin et al., 2001]. For the purpose of this thesis, we describe here a muscle contraction model based on the corotational linear elasticity in the spirit of [Saito et al., 2015].

For a given muscle activation level $\alpha \geq 1 \in \mathbb{R}$, we define a matrix $\mathbf{S}$ as follows:

$$\mathbf{S}(\alpha) = \begin{pmatrix} \alpha^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.10}$$

This matrix is used as a linear muscle activation transformation of every element rotated in a way that aligns the x-axis with a fiber direction. A muscle activation transformation of each element reflecting the activation level in a global frame becomes:

$$\mathbf{A}(\alpha) = \mathbf{B}(\vec{b})\mathbf{S}(\alpha)\mathbf{B}(\vec{b})^T \tag{2.11}$$

where $\mathbf{B} \in SO(3)$ denotes a rotation matrix that aligns an element with its fiber direction $\vec{b}$. To integrate this transformation into a constitutive model, we define a new muscle activation deformation gradient as follows:

$$\mathbf{F}_a(\alpha) = \mathbf{D}(\mathbf{x})(\mathbf{A}(\alpha)\mathbf{D}(\mathbf{X}))^{-1} = \mathbf{D}(\mathbf{x})\mathbf{D}(\mathbf{X})^{-1}\mathbf{A}(\alpha)^{-1} = \mathbf{F}\mathbf{A}(\alpha)^{-1} \tag{2.12}$$

Here we transform the shape matrix (Eq. 2.4) of an undeformed configuration of an element to reflect the muscle activation. This approach is usually used to model plastic deformations. In our case, the transformation is a function of the activation level α. Finally, the corotational muscle activation model is defined as:

$$\mathbf{E}_{muscle}(\mathbf{F}, \alpha) = \sum_{i=1} w_i \mu_i \|\mathbf{F}_a(\alpha) - \mathbf{R}_i\ \|_F^2 + \sum_{i=1} w_i \frac{\lambda_i}{2} tr(\mathbf{R}_i^T \mathbf{F}_i - \mathbf{I})^2 \tag{2.13}$$

Please note that we only apply the activation transformation in the first term of the corotated elasticity. The second term which penalizes resistance to a volume change is not affected by the muscle activation transformation as it reflects the incompressibility of a muscle tissue, i.e., a muscle tissue volume should not change due to a muscle contraction and it is up to a solver to find a solution that satisfies both constraints to a given degree.

2.3 Simulation methods

In this section, we describe how to compute a motion of a deformable object given initial conditions and external forces acting on the object for a specified time period.

In a classical mechanics, a dynamical system like this can be derived from Newton's laws of motion. More specifically, Newton's second law defines an important relationship between force, mass and acceleration which we use to derive equations of motion for our elastic bodies representing soft tissues.

2.3.1 Explicit integration scheme

Given a total force and information about an object's mass distribution, we can compute an acceleration that can be further integrated over time to obtain change of positions (velocities) and hence a motion of an object for one time-step. This approach is called an explicit integration or forward Euler method. The missing part to assemble such system is evaluation of forces.

Previous sections (e.g. Section 2.2.3) described strain energies over tetrahedralized continuum of a body as functions of a deformation gradient $\mathbf{E}(\mathbf{F})$. We can compute force of a jth vertex of a tetrahedral element i as follows:

$$\vec{f}_{i,j} = -\frac{\partial \mathbf{E}(\mathbf{F})}{\partial \mathbf{F}_i}\frac{\partial \mathbf{F}_i}{\partial \vec{x_j}} \tag{2.14}$$

In the case of linear elasticity (Section 2.2.3), the partial derivative of the strain energy with respect to the deformation gradient becomes:

$$\mathbf{P}(\mathbf{F}) = \frac{\partial \mathbf{E}(\mathbf{F})}{\partial \mathbf{F}_i} = w_i\mu_i(\mathbf{F}_i + \mathbf{F}_i^T - 2\mathbf{I}) + w_i\lambda tr(\mathbf{F}_i - \mathbf{I})\mathbf{I} \tag{2.15}$$

This gives all necessary components for the simulator. Unfortunately, the explicit integration can easily become unstable for complex systems like ones we work with in this thesis. The solution of this problem is to use an implicit method (backward Euler) described in the next section.

2.3.2 Backward Euler method as an optimization problem

In this dissertation, we leverage a formulation of backward Euler casted as a minimization problem [Gast et al., 2015]. The main idea of this approach is to define an optimization problem such that its solution satisfies a general system of non-linear equations $g(x) = 0$. In our case, $g(x) = 0$ is a dynamical system we are trying to solve. If we constrain ourselves to using local optimization methods (for performance reasons), we can assume that we can always find just a local optimum. However, we can use the fact that a local optimum of an objective function we would like to optimize is at a point where its gradient is zero. Thus, we are looking for an objective function definition $G(x)$ of our optimization problem such that $g(x) = \frac{\partial G(x)}{\partial x} = 0$, i.e., we are simply looking for an integral of a function $g(x)$. For well-posed problems, this gives us an efficient and well-studied framework where we can always find a solution to our complex physics-based systems up to limits of numerical accuracy.

Quasi-static system optimization

Let us first look at a quasi-static system optimization example where we assume small enough velocities so that they can be ignored. In such system, all internal forces are in equlibrium with external forces, i.e., $f_{int} + f_{ext} = 0$, where internal forces can be elastic forces and external forces can include gravity or soft constrains of a system. In Eq. 2.14, we showed that a force is a negative derivative of a strain energy function. This leads us to an elegant formulation where we can directly use a strain energy function as a term in the objective function.

An example quasi-static elastic system can be formulated as an optimization problem as follows:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \sum_{i=1} \mathbf{E}_{Linear}(\mathbf{F}_i(\mathbf{x}, \mathbf{X})) + \mathbf{E}_{Gravity}(\mathbf{x}) + \mathbf{E}_{Constr}(\mathbf{x}) \tag{2.16}$$

where $\mathbf{x}$ are deformed tetrahedra vertices, $\mathbf{X}$ are undeformed (constant) tetrahedra vertices, $\mathbf{F}_i$ is a deformation gradient of ith tetrahedron, $\mathbf{E}_{Gravity}(\mathbf{x})$ is a gravity potential energy function and $\mathbf{E}_{Constr}(\mathbf{x})$ is a spring potential energy function following Hooke's law.

Dynamic system optimization

Adding dynamics into a system requires nothing more than an additional term in the objective sometimes referred to as a *momentum potential* [Bouaziz et al., 2014] which we describe now.

A dynamic state of our system at a timestamp t is given by vertices $\mathbf{x}_t$ and velocities $\mathbf{v}_t$ of a tetrahedral mesh. Following Newton's second law and Backward Euler integration scheme we get:

$$\mathbf{M}\frac{\mathbf{v}_t - \mathbf{v}_{t-1}}{h} = f_{int}(\mathbf{x}_t) + f_{ext}(\mathbf{x}_t)$$
$$\mathbf{v}_t = \frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{h}$$

where h is a time-step size and $\mathbf{M}$ is a mass matrix. We can further modify the equation by eliminating $\mathbf{v}_t$:

$$\frac{1}{h^2}\mathbf{M}(\mathbf{x}_t - \mathbf{x}_{t-1} - h\mathbf{v}_{t-1}) - (f_{int}(\mathbf{x}_t) + f_{ext}(\mathbf{x}_t)) = 0 \tag{2.17}$$

Finally, casting this equation to an optimization problem yields:

$$\min_{x} \frac{1}{h^2}\|\mathbf{M}^{\frac{1}{2}}(\mathbf{x}_t - \mathbf{x}_{t-1} - h\mathbf{v}_{t-1})\|_F^2 + \mathbf{E}_{int}(\mathbf{x}_t) + \mathbf{E}_{ext}(\mathbf{x}_t) \tag{2.18}$$

Comparing this to the quasi-static example (Eq. 2.16), we simply added an additional term we denote as $\mathbf{E}_{Dynamic}$ to the rest of the optimization problem:

$$\mathbf{E}_{Dynamic} = \frac{1}{h^2}\|\mathbf{M}^{\frac{1}{2}}(\mathbf{x}_t - \mathbf{x}_{t-1} - h\mathbf{v}_{t-1})\|_F^2 \tag{2.19}$$

This makes the simulation framework versatile as it is very easy to switch between quasi-static and dynamic system optimization simply by adding or removing the $\mathbf{E}_{Dynamic}$ term.

2.3.3 Optimization solvers

In this section, we briefly describe methods for solving non-linear continuous optimization problems such as ones we formulated in Section 2.3.2 where only a local solution is required. Without loss of generality, we are only interested in minimization problems. We recommend reffering to [Nocedal and Wright, 2006] for a comprehensive description of optimization algorithms.

Gradient descent

The basic principle of iterative methods such as Gradient descent is to find a descent direction p_i of a given differentiable function $f(x)$ at point x_i and search along the descent direction a new point x_{i+1} such that

$$f(x_i) > f(x_{i+1}) = f(x_i + \alpha_i p_i) \tag{2.20}$$

where α_i is a step size that can be determined by a line search method (see [Nocedal and Wright, 2006]) and $p_i = -\nabla f(x_i)$ in case of Gradient descent. The process typically continues until some termination condition is met. Termination conditions can include a number of iterations $i < N$, norm of the gradient $|\nabla f(x_i)| < \epsilon$ or a value of an objective $f(x_i) < A$.

The performance of optimization algorithms can vary greatly depending on a specific problem. At every iteration the method requires one evaluation of the gradient and a number of objective evaluations for the line search. For elastic terms, this means evaluating forces and a strain energy which we showed in the previous section. While this is a relatively small computational cost per iteration, second-order methods such as Newton's method have typically a superior asymptotic rate of convergence and better amortized computation time for a class of problems we discuss in this thesis despite requiring additional evaluation of a Hessian [Wang and Yang, 2016, Liu et al., 2017].

Newton's method

Newton's method can typically make larger steps than Gradient descent because it is leveraging a second-order Taylor expansion of an objective function. For Newton's method, a descent direction is defined as:

$$p_i = -\frac{\nabla f(x_i)}{\nabla^2 f(x_i)} \tag{2.21}$$

In practice, we do not evaluate an inverse of a Hessian, but solve for p_i in the following linear system:

$$\nabla^2 f(x_i) p_i = -\nabla f(x_i) \tag{2.22}$$

For well-posed problems, a Hessian $\nabla^2 f(x_i)$ is a symmetric positive definite matrix and we can use a Cholesky decomposition [Guennebaud et al., 2010] to solve for the descent direction. In some cases, such as linear elasticity (Section 2.2.3), the approximation by second-order Taylor expansion is a perfect approximation and the solution is found in just one iteration. The Hessian matrix is constant in this case and we can precompute the matrix factorization step to speed up a simulation.

In some situations, we can not guarantee positive definiteness of a Hessian. For example, this can happen when there are not enough constraints or when some tetrahedra are inverted. In case we can not fix such issue locally for each tetrahedron, we can progressively apply Tikhonov regularization on a full Hessian matrix until it is positive definite.

Constrained optimization

Many of the problems we solve in this work are formulated as constrained optimization problems. For equality constraints $c(x) = 0$, we can leverage a method of Lagrange multipliers and solve a new problem defined by the Lagrangian function that contains additional parameters depending on the number of constraints. When converged successfuly, constraints are satisfied with high accuracy.

Another way for solving equality constraints is called a penalty-based method. In this approach, constraints are added as an additional term in an objective as a weighted penalty:

$$\min_x f(x) + \beta c(x)^2 \tag{2.23}$$

The weight β is initially set to a small number and the problem is solved in the same way as an unconstrained one. β is then progressively increased in each solver iteration until constraints or other solver termination criteria are met. The number of iterations, initial weight and the penalty factor increment can differ widely depending on the problem.

2.3.4 Hessian approximation

The main disadvantages of second-order methods are the descent direction solve and a performance of the Hessian matrix evaluation. For most forward problems, Hessian is typically a sparse matrix $\nabla^2 f(x_i) \in \mathbb{R}^{N\times N}$, where N is a number of optimization parameters. In case where no additional constraints are dynamically added to the system, a sparsity pattern is determined by a mesh configuration and remains constant which can be leveraged for a performance boost in some solvers. In general, both direct and iterative sparse linear solvers can be used to evaluate the descent direction depending on the size of a problem and other factors [Guennebaud et al., 2010].

Some constraints especially ones introduced in inverse problems discussed in the next chapter have large dense blocks in the Hessian matrix. Solving a large and a dense system is usually the biggest bottleneck of the overall problem. For this reason, there exist methods that try to iteratively approximate Hessian to avoid computationally intensive evaluation (L-BFGS), and some methods propose suitable constant Hessian which allows pre-factoring the matrix only once in a preprocessing stage [Liu et al., 2017]. This approach is referred to as a Quasi-Newton method [Nocedal and Wright, 2006]. The descent direction evaluation is straighforward in this case:

$$p_i = -\frac{\nabla f(x_i)}{\mathbf{A}} \tag{2.24}$$

where $\mathbf{A} \approx \nabla^2 f(x_i)$. A popular method based on this idea in Computer Graphics for animating deformable objects is called Projective Dynamics [Bouaziz et al., 2014, Liu et al., 2017]. We used this method in forward simulation problems where we achieved real-time performance.

3. Solving inverse problems

3.1 Introduction

A general inverse problem can be defined as an inference of mathematical model parameters based on observations [Vogel, 2002]. In this dissertation, these observations consist of human body and face captures of a person, e.g. surface meshes, under various conditions. In order to fit large variety of data, our models should ideally generalize well enough such that we can explain both variations in each person's shape but also their pose or facial expression. We are therefore interested in two sets of parameters: person-specific parameters that are shared between observations and parameters specific to an observation, such as a skeletal pose, muscle activation or facial expression. Shared parameters for multiple observations are especially important to reduce ambiguity for ill-posed problems that we are working with in this thesis.

Another important factor reducing ambiguity is our anatomical prior that we refer to as a *template anatomical model.* The template model corresponds to an average human body and head shape and includes surface data of a skeleton, skeletal muscles and a skin. In next chapters, we show how we use this prior for a formulation of various growth or fitting models that are able to match diverse characters but we also use the prior as an important regularization of a skeletal system and soft tissues that are not part of surface observations.

Once we determine all parameters of our models, we can run a forward simulation and generate a novel animation with unseen poses or expressions that will hopefully match reality or test dataset not used for model parameter inference for evaluation purposes. We can also change person-specific parameters and for example easily create a more muscular character as shown in [Saito et al., 2015]. Such process of a manual parameter control can be understood as *forward anatomical modeling.* In contrast, we refer to an automated inferrence of person-specific parameters from data such as bone lengths, muscle shape or a fat distribution as *inverse anatomical modeling* which is the main subject of this thesis.

3.2 Optimization formulation of inverse problems

As a toycase example, consider a simulation of a deformable object modeled with an elastic material parameterized by a homogeneous stiffness E and a given mass distribution. Some part of this object can be positionally constrained in space. The rest is moving freely and we are observing effects of gravity force affecting the shape of the object. Changing the stiffness parameter E and running the simulation that outputs new tetrahedral mesh vertices $\mathbf{x}$ is a typical example of a forward problem:

$$\mathbf{x} = f(\mathbf{X}, E) \tag{3.1}$$

where $\mathbf{X}$ are undeformed mesh vertices of a simulated object and f is a simulation function derived from methods described in Chapter 2 of this thesis (see Eq. 2.16

for example).

In an inverse problem, the target observation data $\mathrm{X}_{\mathrm{tgt}}$ can consist of all mesh vertices or just surface vertices representing a deformed object due to gravity with unkown stiffness. The goal is to find a stiffness parameter E such that a forward problem model best matches the input $\mathrm{X}_{\mathrm{tgt}}$. Turning this into an optimization problem gives:

$$\min_{E} \|f(\mathbf{X}, E) - \mathrm{X}_{\mathrm{tgt}}\| \tag{3.2}$$

Solving this problem can be very challenging since an evaluation of the simulation function f requires solving another optimization problem (Eq. 2.16). We call such problems *inverse physics problems*. The following sections cover some of the existing approaches to solving such problems.

3.3 Inverse kinematics

Before we discuss optimization of inverse physics problems, we first describe an inverse problem that illustrates a similar, perhaps more familiar, concept. Inverse kinematics (IK) is very well known in the field of computer animation and robotics [Aristidou et al., 2018] and is also a subproblem in this thesis.

We use kinematics to describe a motion of a skeleton that we define as a chain of transformations. These include translations of joint offsets from parents $\mathbf{T}^{Transl}$ (typically constant) and rotational joints parameterized by Euler angles $\mathbf{T}^{Rot}$. Let us define a transformation from a root to a joint j as follows:

$$\mathbf{T}_{0\to j}(\theta) = \mathbf{T}_j^{Transl}\mathbf{T}_j^{Rot}(\theta_j)\cdots\mathbf{T}_0^{Transl}\mathbf{T}_0^{Rot}(\theta_0) \tag{3.3}$$

And a forward kinematics (FK) process as:

$$\vec{x}_j = \mathbf{FK}_j(\theta, \vec{X}_j) = \mathbf{T}_{0\to j}(\theta)\mathbf{T}_{0\to j}^{-1}(\mathbf{0})\vec{X}_j \tag{3.4}$$

where $\vec{X}_j$ is a position of a point in world-space coordinates attached to a joint j in a neutral pose, $\vec{x}_j$ is a position of a point in a pose defined by θ in world-space coordinates. The inverse transformation term in a neutral pose $\mathbf{T}_{0\to j}^{-1}(\mathbf{0})$ can be precomputed for each joint and allows us to work directly in world-space coordinates. Both affine transformations and points can be represented in homogeneous coordinates to turn the function $\mathbf{FK}_j$ into simple matrix multiplications.

The goal of inverse kinematics is to determine a pose θ given a position of a point attached to a joint $\vec{x}_j$. For small enough skeletal systems, IK can be solved analyically [Aristidou et al., 2018]. However, for more complex systems where more solutions exist or for systems with joint limits, we can cast inverse kinematics as an optimization problem:

$$\theta = \mathbf{IK}_j(\vec{x}_j, \vec{X}_j) = \arg\min_{\theta} \|\mathbf{FK}_j(\theta, \vec{X}_j) - \vec{x}_j\| \tag{3.5}$$

and solve using methods presented in Section 2.3.3. The system can be trivially extended to solve for multiple points simultaneously to reduce ambiguity.

Note that the forward problem function $\mathbf{FK}_j$ is defined explicitly as the transformation chain and its gradient $\nabla_\theta \mathbf{FK}_j$ and potentially even Hessian required for the inverse problem optimization can be derived analytically.

3.4 Adjoint method

In this section, we show how to solve an inverse problem in case where a forward problem can not be expressed explicitly, such as a simulation of a deformable object, and it is therefore more complicated to express derivatives necessary for numerical solvers as opposed to the example in the previous section.

Let us continue with a toycase example described in Section 3.2 where the inverse problem is to find stiffness parameters $E \in \mathbb{R}^m$ for a given observation $\mathrm{X}_{\mathrm{tgt}}$ and a set of constraints. To make the formulation concise, we define a forward function as a following quasi-static problem:

$$\mathcal{X}(E) = \arg\min_{\mathbf{x}} \mathbf{Q}(\mathbf{x}, E) \tag{3.6}$$

$$\mathbf{Q}(\mathbf{x}, E) = \sum_{i=1} \mathbf{E}_{Elastic}(\mathbf{F}_i(\mathbf{x}, \mathbf{X}), E) + \mathbf{E}_{Gravity}(\mathbf{x}) + \mathbf{E}_{Constr}(\mathbf{x}) \tag{3.7}$$

where $\mathbf{X} \in \mathbb{R}^{3n}$ are given constant undeformed mesh vertices, $\mathbf{x} \in \mathbb{R}^{3n}$ are deformed vertices, $\mathbf{E}_{Elastic}$ is arbitrary elastic energy parameterized by a deformation gradient $\mathbf{F}_i$ and stiffness E.

An inverse problem $\mathbf{G}$ we wish to solve is then defined as:

$$\min_E \mathbf{G}(E) = \frac{1}{2}\|\mathcal{X}(E) - \mathrm{X}_{\mathrm{tgt}}\|^2, \tag{3.8}$$

i.e., we are trying to find E that would best explain observations $\mathrm{X}_{\mathrm{tgt}}$. The gradient required for a numerical solver is:

$$\nabla_E \mathbf{G} = \nabla_{\mathcal{X}} \mathbf{G} \nabla_E \mathcal{X} = (\mathcal{X}(E) - \mathrm{X}_{\mathrm{tgt}})^T \nabla_E \mathcal{X} \in \mathbb{R}^{1 \times m}, \tag{3.9}$$

the challenging term being $\nabla_E \mathcal{X} \in \mathbb{R}^{3n \times m}$.

The solution to the forward problem $\mathcal{X}$ defined above is at a quasi-static equlibrium where sum of forces $q \in \mathbb{R}^{1 \times 3n}$ are zero:

$$q(\mathbf{x}, E) = \nabla_x \mathbf{Q}(\mathbf{x}, E) = 0 \tag{3.10}$$

which implies:

$$q(\mathcal{X}(E), E) = \nabla_{\mathcal{X}} \mathbf{Q}(\mathcal{X}(E), E) = 0 \tag{3.11}$$

Next, in order to express $\nabla_E \mathcal{X}$, we differentiate the last equation by E and obtain:

$$\nabla_E q(\mathcal{X}(E), E) = 0 \tag{3.12}$$

$$\nabla_{\mathcal{X}} q \nabla_E \mathcal{X} + \nabla_E q = 0 \tag{3.13}$$

$$\nabla_{\mathcal{X}\mathcal{X}} \mathbf{Q} \nabla_E \mathcal{X} + \nabla_{\mathcal{X}E} \mathbf{Q} = 0 \tag{3.14}$$

$$\nabla_E \mathcal{X} = -(\nabla_{\mathcal{X}\mathcal{X}} \mathbf{Q})^{-1} \nabla_{\mathcal{X}E} \mathbf{Q} \tag{3.15}$$

This gives us a linear system with variables $\nabla_E \mathcal{X}$.

We can increase performance by directly evaluating $\nabla_E \mathbf{G}$. We multiply Eq. 3.15 by $\nabla_\chi \mathbf{G}$ from the left:

$$\nabla_\chi \mathbf{G} \nabla_E \chi = -\nabla_\chi \mathbf{G} (\nabla_{\chi\chi} \mathbf{Q})^{-1} \nabla_{\chi E} \mathbf{Q} \tag{3.16}$$

and let

$$\lambda = -\nabla_\chi \mathbf{G} (\nabla_{\chi\chi} \mathbf{Q})^{-1} \tag{3.17}$$

This gives a new linear system where $\lambda \in \mathbb{R}^{1\times 3n}$. The gradient of the inverse problem $\mathbf{G}$ becomes:

$$\nabla_E \mathbf{G} = \lambda^T \nabla_{\chi E} \mathbf{Q} \tag{3.18}$$

This method is referred to as the adjoint method and λ is called the vector of adjoint variables. Please refer to Kriz and Pultr [2013], Bradley [2013], Sifakis et al. [2005] for additional details and background in mathematical analysis.

3.5 Equilibrium constraint method

A different approach to solving an inverse physics problem is to reformulate the optimization problem by introducing variables representing the quasi-static solution that are coupled with the forward quasi-static function by a constraint.

We are following the same example inverse problem as in Section 3.4. Formally, we reformulate the inverse problem $\mathbf{G}$ as follows:

$$\begin{aligned} \min_{\xi=\{\mathbf{x},E\}} \mathbf{G}(\xi) &= \frac{1}{2}\|\mathbf{x} - \mathrm{X}_{\mathrm{tgt}}\|^2 \\ \text{subject to } \nabla_{\mathbf{x}} \mathbf{Q}(\mathbf{x}, E) &= 0 \end{aligned} \tag{3.19}$$

where Q is a total energy of the quasi-static system defined in Eq. 3.7 and $\nabla_{\mathbf{x}} \mathbf{Q}(\mathbf{x}, E) \in \mathbb{R}^{1\times 3n}$.

The main difference to the formulation described in Section 3.4 is that Eq. 3.19 does not contain implicitly defined functions that require solving another optimization problem. This makes expressing derivatives of the inverse problem more straighforward.

One might be naively tempted to change the constraint in Eq. 3.19 to $\mathbf{Q}(\mathbf{x}, E) = 0$ in order to avoid 3rd-order derivatives of the scalar-valued energy function. However, such formulation is incorrect and does not solve our inverse problem since we are not interested in zero-energy solution, but rather a quasi-static equlibrium where sum of all forces are zero, i.e., $\nabla_{\mathbf{x}} \mathbf{Q}(\mathbf{x}, E) = 0$. Depending on the specific problem, the solution can have an arbitrary energy $\mathbf{Q}(\mathbf{x}, E) > 0$.

3.5.1 Hessian approximation of the constraint

We can formulate the Eq. 3.19 as an unconstrained problem by using the penalty-based method described in Section 2.3.3:

$$\min_{\xi=\{\mathbf{x},E\}} \mathbf{G}(\xi) = \frac{1}{2}\|\mathbf{x} - \mathrm{X}_{\mathrm{tgt}}\|^2 + \frac{\beta}{2}\|\nabla_{\mathbf{x}} \mathbf{Q}(\mathbf{x}, E)\|^2 \tag{3.20}$$

where β is a penalty weight. Solving this problem by the Newton method requires evaluation of both gradient $\nabla_\xi \mathbf{G}$ and Hessian $\nabla^2_\xi \mathbf{G}$.

The first derivative of the penalty term with respect to parameters ξ is:

$$\frac{\partial \frac{\beta}{2} \|\nabla_{\mathbf{x}} \mathbf{Q}(\mathbf{x}, E)\|^2}{\partial \xi} = \beta \frac{\partial \mathbf{Q}}{\partial \mathbf{x} \partial \xi}^T \frac{\partial \mathbf{Q}}{\partial \mathbf{x}} \in \mathbb{R}^{|\xi| \times 1} \tag{3.21}$$

where $|\xi| = 3n + m$.

The second derivative of the penalty term is then:

$$\frac{\partial (\beta \frac{\partial \mathbf{Q}}{\partial \mathbf{x} \xi}^T \frac{\partial \mathbf{Q}}{\partial \mathbf{x}})}{\partial \xi} = \beta \frac{\partial \mathbf{Q}}{\partial \mathbf{x} \partial \xi}^T \frac{\partial \mathbf{Q}}{\partial \mathbf{x} \partial \xi} + \beta \frac{\partial \mathbf{Q}}{\partial \mathbf{x} \partial \xi \partial \xi}^T \frac{\partial \mathbf{Q}}{\partial \mathbf{x}} \tag{3.22}$$

As indicated before, the term in the Hessian matrix $\frac{\partial \mathbf{Q}}{\partial \mathbf{x} \partial \xi \partial \xi} \in \mathbb{R}^{3n \times |\xi| \times |\xi|}$ contains third-order derivatives of energy functions of $\mathbf{Q}$ defined in Eq. 3.7 with respect to $\mathbf{x}$. In practice, these derivatives are often negligible and are typicaly ignored Bickel et al. [2012].

4. Inverse Human Body Modeling

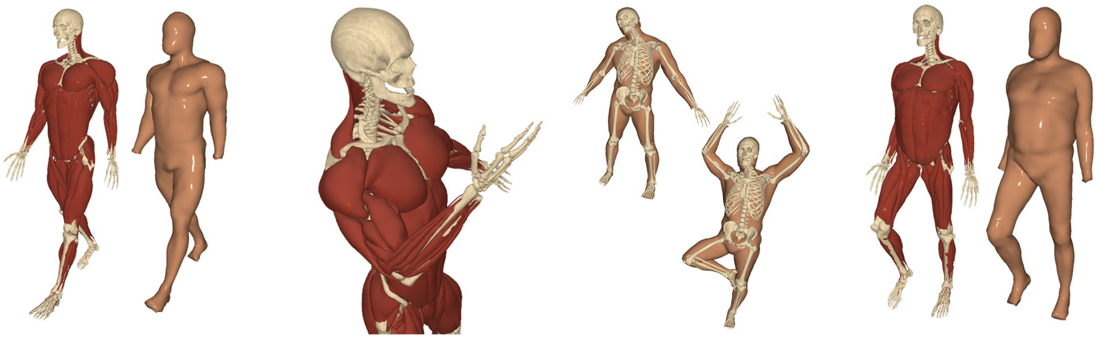


Figure 4.1: We present a full-body reconstruction and animation system that can simulate physics-based volumetric effects such as self-collision and inertial effects. Our method uses a set of 3D surface scans to adapt an anatomically-inspired volumetric model to the user.

This chapter is based on the publication Kadlecek et al. [2016]. The first authorship for this paper has been shared with Alexandru Eugen Ichim with whom the author collaborated closely. They both implemented roughly half of the code required for this project. The author mainly contributed by experiments and implementation of the following:

- forward skeleton kinematic model
- inverse skeleton adaptation and bone growth models
- forward physics-based skinning model
- rest-pose optimization terms of the inverse problem
- body model registration pipeline
- symmetric as-rigid-as-possible energy formulation important for convergence of the inverse problem

4.1 Introduction

The importance of human anatomy in visual arts was appreciated already by Renaissance masters such as Leonardo da Vinci. More recently, 3D anatomical models combined with physics-based simulation have been used to deliver unprecedented visual realism in modern computer generated movies. Unfortunately, the design of anatomically realistic characters is a labor intensive process even for experienced digital artists using professional modeling and simulation tools, such as those developed at Weta Digital and the ILM. Therefore, high-fidelity

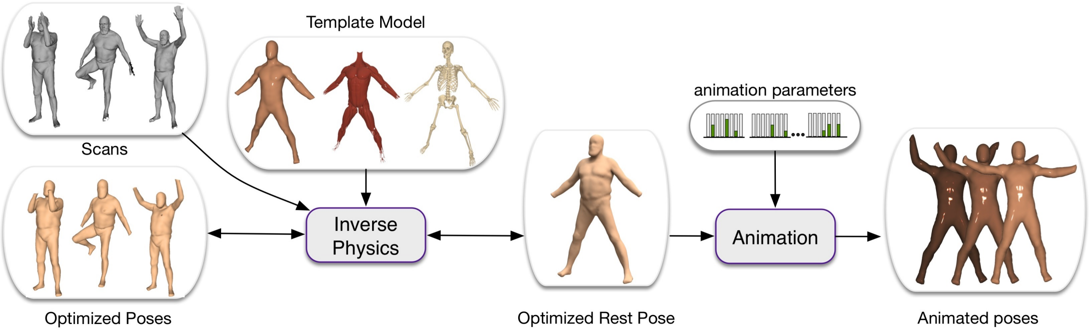


Figure 4.2: Workflow of our method: We take as input a set of 3D scans of the same actor in different poses. Our method aims at reconstructing a complete volumetric, rigged, and physics-ready body model of the actor, by starting from an anatomical template model of an average male. This consists of extracting its exterior and interior shapes, as well as skeleton bone lengths. Finally, our models are ready to be animated using external skeletal and muscle activation data.

anatomical models are typically only affordable in high-budget production, e.g., in movies such as Avatar or The Lord of The Rings trilogy. Even though modeling of imaginary creatures such as dragons inherently relies on creativity of digital artists, when it comes to modeling humans, we believe we can substantially improve upon the state of the art.

In this chapter, we present an automatic method to create an anatomical, physics-based model of the body of a given human subject, e.g., an actor (see Figure 4.1). We achieve this by capturing a set of full-body 3D scans in various poses and combining it with a template anatomical model. This template model represents the anatomy of an average human body, similar to traditional medical atlases. However, actual human bodies exhibit large variations in height, muscularity, adiposity, proportions of the limbs, etc. Our goal is to reshape and rescale the template anatomical model in order to fit the target scans as closely as possible, while accounting for shape changes due to both subject-specific variations (bone lengths, muscularity, adiposity, ...) as well as due to posing (changes of joint angles). The first type of deformations (subject-specific) are caused by long-term biological growth processes, while the pose-based deformations are induced by short-term voluntary muscle contractions and consequent joint motion. Our approach is summarized in Figure 4.2.

Data-driven modeling of animated human bodies has been a long standing topic in computer graphics. Systems such as SCAPE Anguelov et al. [2005] or the more recent BlendSCAPE Hirshberg et al. [2012] construct an articulated human body model from a set of input 3D scans. Similarly to artist-directed systems such as Pose Space Deformation Lewis et al. [2000], these methods build a data-driven model which predicts skin deformations based on the skeletal pose (i.e., joint rotations). However, these methods focus exclusively on the skin, i.e., outer boundary of the body. The skeleton is modeled as connected line segments, disregarding the volumetric nature of bones or muscles. While surface-based data-driven methods are effective at interpolating the input scans, they are oblivious to the fact that biological soft tissues are elastic solids subject to Newton's laws of motion. A notable exception is DYNA Pons-Moll et al. [2015], which we will discuss in Related Work. To our knowledge, our method is the first to reconstruct a fully physics-based subject-specific anatomical model, naturally supporting effects such as inertia, collisions, and gravity. We found that already volumetric modeling of organs and their corresponding stiffness has interesting visual implications; e.g., the rigidity of the rib cage is clearly visible when animating upper trunk rotations, such as in Figure 4.8.

The problem of reconstructing anatomical models only from surface 3D scans is inherently ill-posed. Ground truth measurements of organs could be obtained using MRI or CT scans; however, these are expensive medical-grade devices designed to diagnose fine-scale pathologies such as bone fractures or tumors. Aside from the high costs, MRI or CT scanners are not suitable for computer animation purposes because they offer only a very limited workspace, i.e., the motion of the imaged human subject is highly constrained; furthermore, MRI machines require long scanning times and CT scanners expose the person to ionizing radiation. Fortunately, for computer graphics purposes we do not need high-fidelity medical imaging, because a rough estimate of the scale and shape of the bones, muscles, and subcutaneous adipose tissues is sufficient to produce high quality animations. Our anatomical model is designed for full-body animations and contains only the most visually significant muscles; we do not model the delicate muscles of the face, hands, and feet, as these body parts are often animated by specialized techniques. Our anatomical template also does not contain the nervous or circulatory systems or models of internal organs. However, our results can readily be combined with other computer graphics techniques such as displacement mapping in order to model, e.g., prominent veins or fine scale wrinkles.

By measuring only the 3D geometry of the skin, it seems impossible to determine what are the shapes and sizes of the underlying bones, muscles, and adipose tissues. However, bones and muscles do not grow arbitrarily in healthy human subjects (we do not consider pathologies in this work), because the musculoskeletal apparatus must be a functional mechanical system to allow locomotion. To quantify which shapes are more likely than others, we employ biomechanics-based growth models similar to Computational Bodybuilding Saito et al. [2015]. While Computational Bodybuilding presented methods for the *forward* simulation of growth of bones, muscles, and adipose tissues, in this work, we study the *inverse* problem, i.e., we formulate an optimization to recover the fitting parameters which best explain our input 3D scans. This problem is quite challenging because we have to account for 1) the fact that each 3D scan is in a different pose and 2) the organs do not grow independently, but influence each other due to internal action-reaction forces (when one bone/muscle grows, it pushes the adjacent organs).

4.1.1 Contributions

To our knowledge, the problem of reconstructing physics-based anatomical models from input 3D scans has not been tackled in previous work. Our main contribution is inverse body modeling (Section 4.5), i.e., formulating and solving a large optimization problem to find a subject-specific anatomical model which explains the input 3D scans as closely as possible. Some parts of our forward skinning model (Section 4.4) are derived from previous work; however, we devise a new elastic potential (which we call “symmetric as-rigid-as-possible” energy) in order to perform the subsequent inverse modeling, since classical as-rigid-as-possible models Sorkine and Alexa [2007] do not work, as we discuss in Section 4.4. We hope that our approach will help to lower the costs of creating anatomical models of humans and make high-quality physics-based animation accessible not only to well-known VFX studios, but to a larger body of researchers and artists.

4.2 Related work

Data-driven techniques. The most common approaches for modeling complex anatomical variation is by leveraging large amounts of data, usually in the form of 3D body scans or performance capture data. Anguelov et al. [Anguelov et al., 2005] learn a statistical model for body shape variations as a function of body pose changes, which is applied on top of a statistical model of neutral-pose body shapes. As such, the same deformation model is used for all the people, while we have the advantage of constructing person-specific internal components which will behave differently in animations. This data-driven approach was extended and applied to sparse motion-capture animation by Loper et al. [Loper et al., 2014], in order to obtain better quality motion reconstructions as compared to traditional skeleton-driven skinning approaches. Zuffi et al. [Zuffi and Black, 2015] propose a part-based model where each body component is a mesh associated to a statistical space, and connected together by pairwise stitching energies. Recently, Pons-Moll et al. [Pons-Moll et al., 2015] introduced a data-driven technique that additionally encodes shape changes due to skin and limb velocity and acceleration, producing animations with compelling inertial effects without the need for a physics simulation. While these techniques are powerful interpolation tools, they are limited in their extrapolation capabilities, fixable only by collecting more and more data. In contrast, our method produces fully physics-based models, naturally supporting not only inertial effects, but also effects due to gravity, volumetric bones, and collisions.

For the particular task of breathing simulation, Tsoli et al. [Tsoli et al., 2014] introduce a data-driven approach in which pose and shape variation is extracted from a set of registered 3D scans of people captured while breathing in different scripted ways. These priors are then used to generate varying types of respiration motions in novel characters. In our method we do not explain shape variations due to breathing, even though this would be an interesting direction for future work.

Anatomical models and physics. The motion of humans and interactions between the various anatomical elements have long been an important focus point for the biomechanics community. OpenSim Delp et al. [2007] is an example of an open-source software framework for biomechanical modeling, simulation and analysis, extensively used in biomechanics and motor control science. However, OpenSIM does not support physics-based volumetric modeling of muscles or adipose tissues. There are also other specialized medically oriented frameworks, such as Sofa Allard et al. [2007] or ArtiSynth Lloyd et al. [2012].

The survey of Lee et al. [Lee et al., 2010] offers a thorough overview of how the biomechanics and computer graphics communities model and simulate muscles, with most work being focused on skeletal muscles. Muscles are very complex structures that are not completely understood by modern medicine, and, as a result, various approximations have been proposed for making muscle simulation tractable for various medical or entertainment applications. Out of those, the physics-based and data-driven approaches are the ones of most interest for our work. Teran et al. [Teran et al., 2003, 2005b,a] introduced some of the first comprehensive approaches for biomechanical human body simulation in computer graphics. They construct a complete volumetric human body and a compatible FEM simulation by using solely data from the Visible Human Dataset.

Saito et al. [Saito et al., 2015] propose a novel system for performing bodybuilding or weight loss simulations on human models. They model the muscles using synthetically computed muscle fibers. The growth of the muscles is discretized into the anisotropic stretch of individual muscle tetrahedra in the direction of the fibers, and computed efficiently using the projective dynamics solver Bouaziz et al. [2014]. The key difference from our method is that Saito et al. [Saito et al., 2015] require the bone/muscle/fat fitting parameters to be provided by the user.

Fan et al. [Fan et al., 2014] propose a framework for simulating a dynamic volumetric musculoskeletal system using an Eulerian-on-Lagrangian discretization that can handle sliding elements in close contact, volume preservation and large deformations.

Anatomy Transfer Dicko et al. [2013] is a method for transferring and editing the internal structure of human bodies. It uses a template human body model containing the skeleton and internal organs and registers it to a single surface-mesh humanoid model. The internal volume is adapted using harmonic deformation, driven by the registration of the exterior surface. The amount of fat tissue is controlled manually and the growth of the bones is constrained for more plausible results. In a similar vein, Zhu et al. [2015] adapts the bone structure of upper and lower limbs given an RGB-D sequence of moving limbs. Comparisons and other differences are discussed in Section 4.8.

While a lot of research has gone into tackling the general problem of human body motion, there has been work targeting specific aspects. For example, Si et al. [Si et al., 2014] use an anatomical body model with muscle actuations in a complex fluid simulation in order to build a control system to simulate different styles of swimming. Similarly, Lee et al. [Lee and Terzopoulos, 2006] focus on the biomechanical modeling and neuromuscular control of the neck region.

Combining simulation and data. A technique for modeling non-linear material deformations from a set of captured examples is introduced by Bickel et al. [Bickel et al., 2009]. They used a scattered data interpolation technique in strain-space to simulate novel deformations of objects composed of the observed materials. Similarly, Wang et al. [Wang et al., 2015] use off-the-shelf 3D sensors to track and model deformations of soft objects using physics-based probabilistic priors. Chen et al. [Chen et al., 2014] propose a performant approach to reconstruct the zero-gravity rest pose shape of an object given multiple observations under various external forces such as gravity.

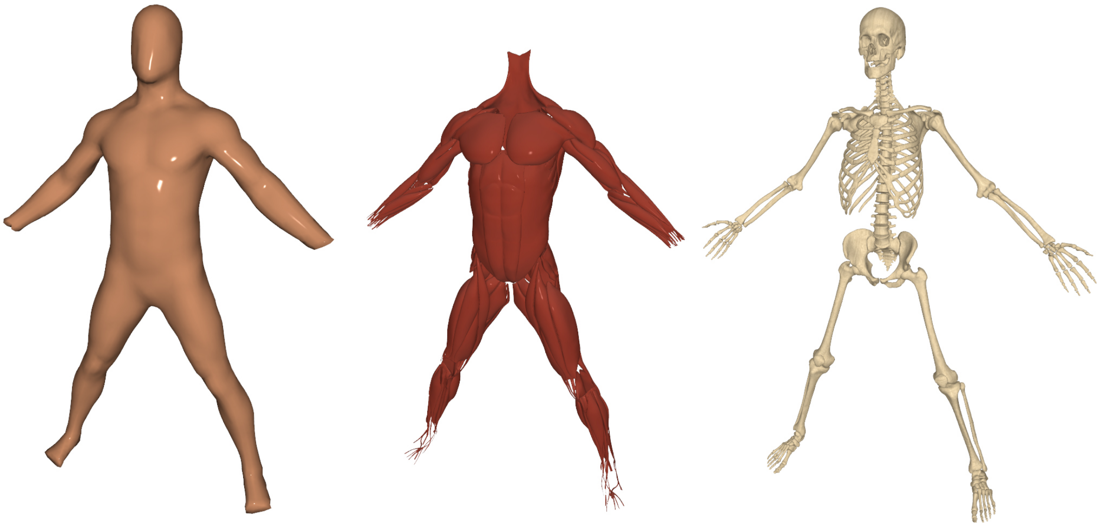


Figure 4.3: Components of our anatomically-inspired volumetric template model. From left to right: skin and underlying generic soft tissue, muscles and tendons, skeleton.

4.3 Template Body Model

The template model defines the topology of the fitted actors, and acts as a regularizer in the reconstruction process (see Fig. 4.3). It consists of a set of n vertices $\mathbf{X}^{\text{tmpl}} \in \mathbb{R}^{3n}$, connected in a tetrahedral mesh. We build the template similar in spirit to Saito et al. [Saito et al., 2015] by starting from the commercially available Zygote anatomical model Zygote [2016] with 111 muscles and 204 bones represented as meshes. The skin, muscles, and bones are uniformly remeshed with the Instant Meshes algorithm Jakob et al. [2015] and then the surfaces are tetrahedralized using the approach of Jacobson et al. [Jacobson et al., 2013].

In our work, we differentiate between four main types of materials: bones, tendons, muscles, and generic soft tissue. Each bone, tendon, and muscle is embedded into the template tetrahedral mesh in a non-conforming way; i.e., each tetrahedron might contain one or all of the materials in certain percentages. These percentages are computed as a pre-processing stage using a Monte Carlo sampling

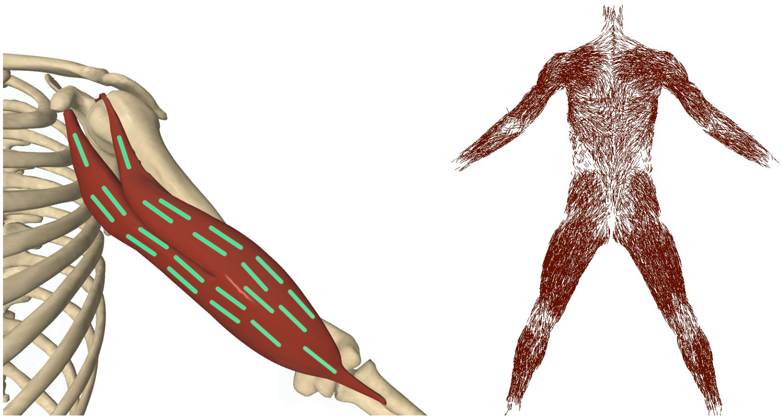


Figure 4.4: Left: a close-up on the fibers on the right biceps muscle. Right: Visualization of the embedded muscle fibers in the template model.

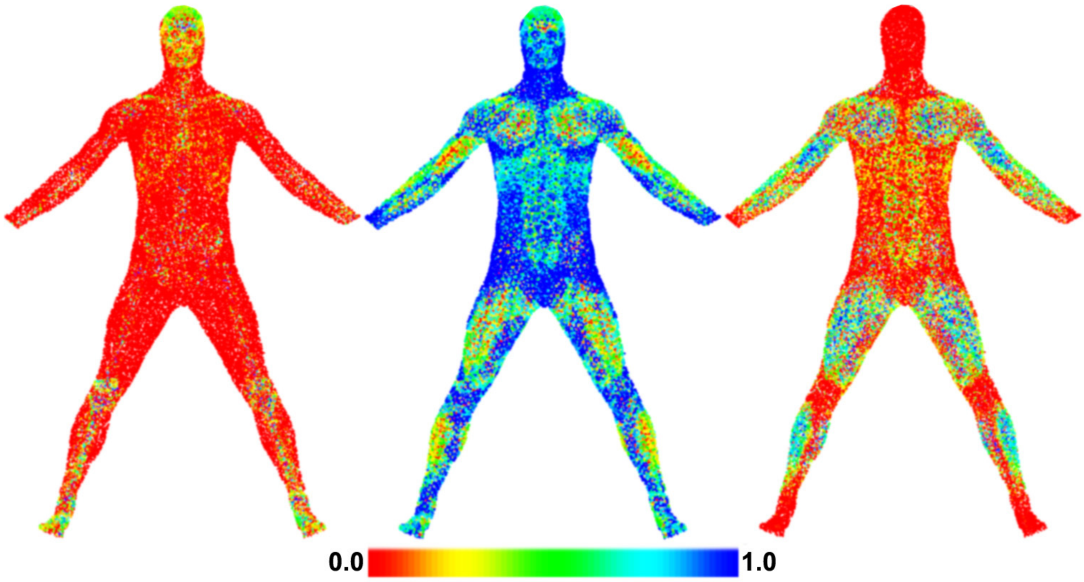


Figure 4.5: The distribution of the material types inside the body. From left to right: bones, generic soft tissue, muscle.

approach to estimate the amount of overlap of each muscle/tendon/bone with each tetrahedron. For modeling the muscle atrophy and hypertrophy during subject-specific body fitting, as well as muscle activations during the animation stage (Section 4.7), the muscle fiber directions are required (see Figure 4.4). We compute the fiber directions in a similar way as Saito et al. [Saito et al., 2015]. First, the tendon regions are selected manually and associated with Dirichlet boundary conditions. The non-tendon muscle boundaries are associated with Neumann boundary conditions. Next, we solve a Poisson equation for a scalar field using these boundary conditions. The resulting muscle fiber directions are aligned with gradients of this scalar field.

Our template anatomical model corresponds to a lean male. To be able to realistically model subjects with larger amounts of subcutaneous fat, we enhance our discretized volumetric template with a “muscle envelope,” Saito et al. [2015], i.e., a triangle mesh which wraps all of the muscles and separates them from the subcutaneous tissues. See Figure 4.5 for a visualization of the material distribution in the template model.

In addition to modeling soft tissue, we also use a realistic skeletal rig to parameterize the allowed motion of the bones. We built our rig using kinematic models established in biomechanics Wu et al. [2002]. The final rig is sufficiently expressive to allow even for complex poses, as shown in Figure 4.6. Also, our rig describes not only pose-dependent variations (via the joint rotation angles $\boldsymbol{\theta}$), but also subject-specific variations (via scaling parameters $\boldsymbol{\pi}$). The scaling parameters $\boldsymbol{\pi}$ allow us to model different lengths and sizes of the bones between individuals. We shall denote $\mathrm{Rig}(\boldsymbol{\theta}_i, \boldsymbol{\pi})$ as the function that describes the motion of the bones as a function of rig parameters. Specifically, the function $\mathrm{Rig}(\boldsymbol{\theta}_i, \boldsymbol{\pi})$ returns posed (skinned) vertex samples, illustrated in Figure 4.9, in the current pose and scaling of the skeletal rig. These vertex samples will be used as boundary conditions for minimizing the elastic energies of the soft tissues, as described below.

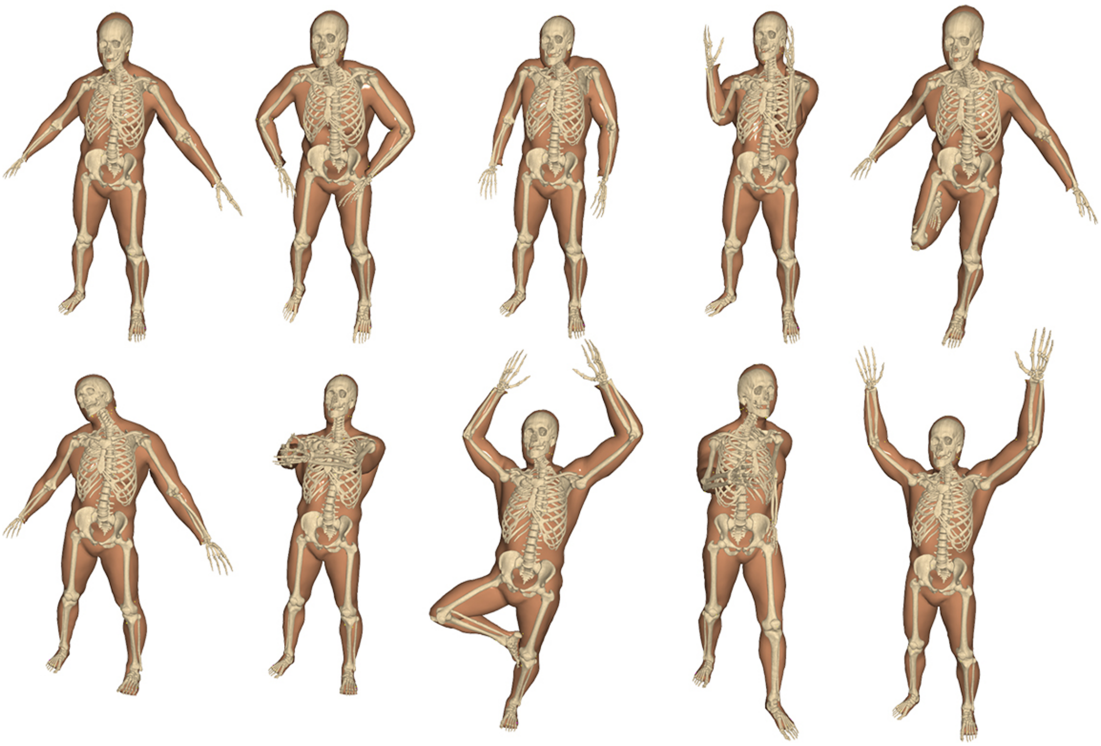


Figure 4.6: Complex skeleton rig fitting on Faust dataset.

4.4 Forward Skinning Model

Before diving into the *inverse* problem of body reconstruction, we first describe our *forward* physics-based character model. Our model is built by extending recent works, in particular Saito et al. [Saito et al., 2015] and Zhu et al. [Zhu et al., 2015]. Saito et al. simulated growth only in the rest pose, without the use of a skeletal rig. Zhu et al. did create a skeletal rig, but only for the extremities (the arm and the leg) and the deformation model was based on direct skinning models.

In this work, the body shape is implicitly defined as minimizer of a deformation energy (corresponding to elasticity of soft biological tissues) subject to Dirichlet boundary conditions (corresponding to the bones which are fixed in a given position in space). This process is known as *quasi-statics* McAdams et al. [2011b]: the bones are kinematically controlled, e.g., by an animator, and for each configuration of the bones, we compute a quasi-static equilibrium where the forces due to bone contacts cancel forces due to internal elasticity of the flesh (we use the term "flesh" as a shorthand for soft biological tissues). These two interpretations are equivalent because forces are negative derivatives of the elastic potential and therefore must be zero in a minimizer.

In equations, we can define the quasi-static solution as function:

$$\mathrm{Skin}(\mathbf{X}^{\mathrm{src}}, \boldsymbol{\theta}_i, \boldsymbol{\pi}) = \arg\min_{\mathbf{X}} E_{\mathrm{skin}}(\mathbf{X}^{\mathrm{src}}, \mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\pi}), \tag{4.1}$$

where $E_{\mathrm{skin}}(\mathbf{X}^{\mathrm{src}}, \mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\pi})$ is equal to the following sum:

$$\mathrm{BoneFlesh}(\mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\pi}) + E_{\mathrm{def}}(\mathbf{X}^{\mathrm{src}}, \mathbf{X}) + E_{\mathrm{grav}}(\mathbf{X}) + E_{\mathrm{col}}(\mathbf{X}). \tag{4.2}$$

Here $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ are joint orientations and bone scaling parameters as discussed in Section 4.3. The vector $\mathbf{X}^{\mathrm{src}}$ describes positions of mesh vertices in a reference *rest*

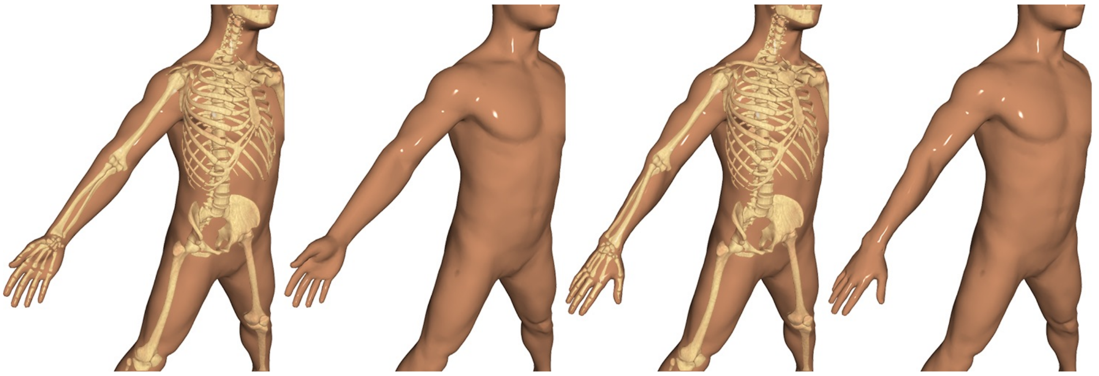


Figure 4.7: Complex pronation-supination motion is handled well by our physics skinning.

pose, while $\mathbf{X}$ corresponds to the *deformed pose*. The BoneFlesh function describes the connection between the deformable mesh representing the flesh and the fixed bones. $E_{\text{def}}(\mathbf{X}^{\text{src}}, \mathbf{X})$ is an elastic potential function which measures the amount of deformation between configurations $\mathbf{X}^{\text{src}}$ and $\mathbf{X}$ (both of which correspond to meshes with the same connectivity). $E_{\text{grav}}(\mathbf{X})$ is the gravity potential, i.e., a linear function which corresponds to the familiar mgh product (mass, gravity constant, height). The gravity potential allows us to simulate the interplay between inertial and gravity forces in a physically realistic way, which is important, e.g., in animating a fat man jumping. Finally, $E_{\text{col}}(\mathbf{X})$ is energy potential penalizing collisions, i.e., self-intersections of the mesh. The necessary condition for $\mathbf{X}$ being in quasi-static equilibrium is $\nabla_{\mathbf{X}} E_{\text{skin}} = 0$, i.e., sum of forces is zero. More details on the above mentioned terms follow.

BoneFlesh. The BoneFlesh term models coupling between kinematically controlled bones and physically simulated flesh. Anatomically, this term can be related to connective tissues which hold the musculoskeletal system together. Mathematically, we define:

$$\text{BoneFlesh}(\mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\pi}) = w_{\text{bone}} \left\| \mathbf{S}^{\text{bone}} \mathbf{X} - \text{Rig}(\boldsymbol{\theta}_i, \boldsymbol{\pi}) \right\|^2, \tag{4.3}$$

where $\mathbf{S}^{\text{bone}}$ is a binary selector matrix which extracts vertices corresponding to the bone vertices kinematically controlled by the Rig function, see Figure 4.9. These vertices are chosen to approximately uniformly sample the surface of the bones and are explicitly present in the tet-mesh associated with $\mathbf{X}$ (conforming embedding). In theory, barycentric (non-conforming) embedding of bone vertices should be sufficient; however, we observed occasional numerical stability issues when nearly co-linear or co-planar vertex samples shared the same tetrahedron. Switching to conforming embedding of bone-samples successfully prevents these issues. For that we use TetGen with a switch to insert additional points Si [2015]. The weighting w_{bone} controls the stiffness of the bone-flesh attachments and is chosen sufficiently high to avoid excessive sliding of the flesh (we note that some sliding is natural because biological connective tissues are compliant). This model is sufficient even for large deformations of the flesh such as pronation/supination (Figure 4.7) or upper trunk rotation (Figure 4.8).

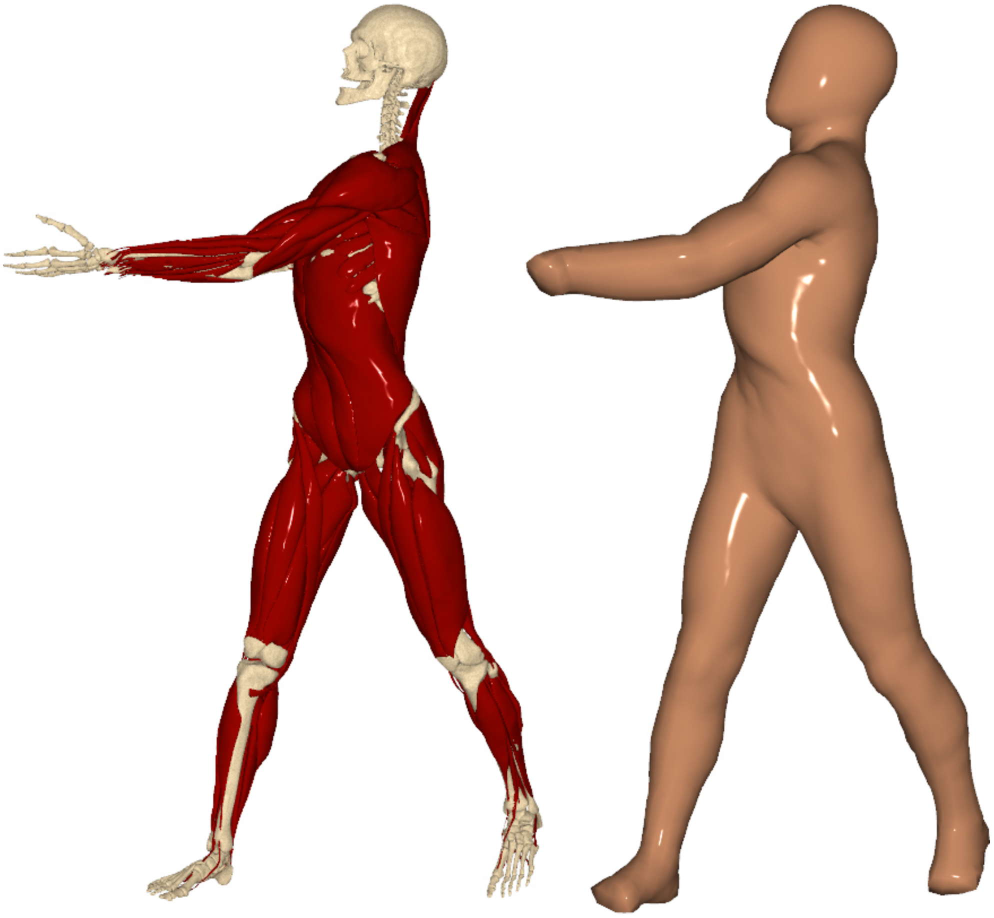


Figure 4.8: Anatomically correct bones produce more realistic body shapes e.g. during upper trunk rotation, where the rib cage retains its shape.

Rig. Our kinematic skeleton is modeled by the function $\text{Rig}(\boldsymbol{\theta}, \boldsymbol{\pi})$, which takes joint angle orientations $\boldsymbol{\theta}$ and bone scaling parameters $\boldsymbol{\pi}$ as input, and produces world-space coordinates of vertices sampling the surfaces of the bones, as shown in Figure 4.9. The Rig function performs two main tasks: 1) it geometrically deforms the bones according to the scaling parameters, allowing us to model individuals with various lengths and shapes of the bones; 2) it implements standard forward kinematics, i.e., hierarchical composition of rotations of individual joints. We currently support only rotational joints, but more complicated joint types (e.g. spline joints Lee and Terzopoulos [2008]) could be added to improve the accuracy of the kinematic modeling.

When changing the lengths and shapes of the bones, it is important not to distort the shape of the bone heads, because adjacent bone heads are often in close sliding contact. We achieve this in a similar way as Zhu et al. [Zhu et al., 2015]. Specifically, each bone is deformed using linear blend skinning with bounded biharmonic weights Jacobson et al. [2011] with handles located in the center of each of the bone heads, see Figure 4.10 for an example of a long bone elongation. The handles of adjacent bones (i.e., forming a joint) are constrained to be transformed by the same matrix which contains only translation and uniform scale. This guarantees that the structure of the joint will be preserved. Bones with more complex shapes, such as the ribcage and spine, are treated in a similar fashion.

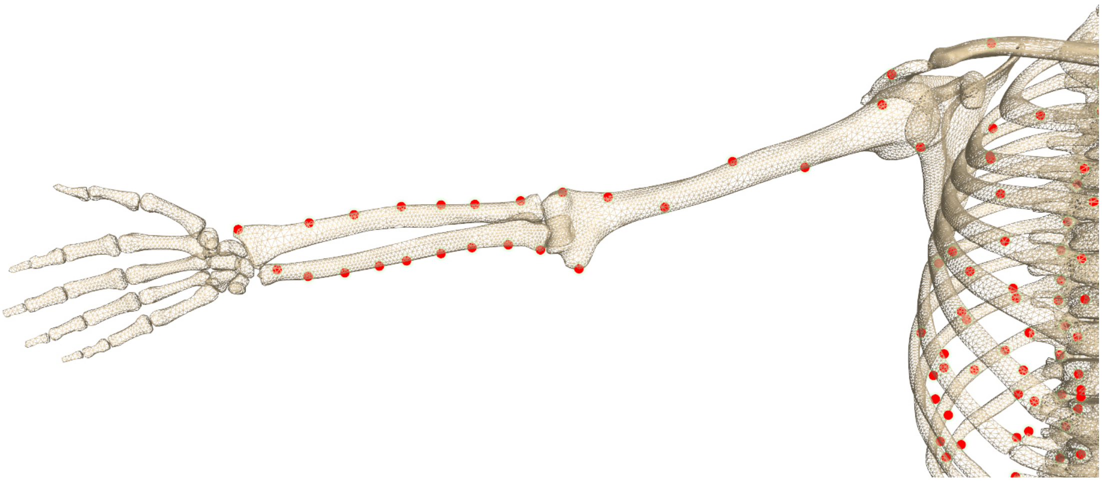


Figure 4.9: Sampled bone vertices corresponding to the selector matrix $\mathbf{S}^{\text{bone}}$ used in BoneFlesh function.

We allow uniform scaling and elongation in the direction of the spinal cord to preserve vertebrae connections. Similarly, we allow elongation of the Scapula in the direction from the Sternum to the Humerus. For every bone, we choose an elongation direction, precompute weight for each sampled bone vertex and set bounds of the parameterization based on anatomical limits. We found that this parameterization was expressive enough to fit all tested scans well. Formally, we can express this deformation using the $\text{BoneFit}(\boldsymbol{\pi})$ function which depends only on the fitting parameters $\boldsymbol{\pi}$ and produces the modified rest pose bone vertex samples as a weighted linear combination of scaling and elongation transformations $\mathbf{T}^{\text{elng}}$:

$$\text{BoneFit}(\boldsymbol{\pi}) = \pi_{scale}\mathbf{S}^{\text{bone}}\mathbf{X}^{\text{src}} + \boldsymbol{\pi}_{elng}\mathbf{W}_{\text{BBW}}\mathbf{T}^{\text{elng}}\mathbf{S}^{\text{bone}}\mathbf{X}^{\text{src}} \tag{4.4}$$

where $\mathbf{W}_{\text{BBW}}$ are precomputed weights, π_{scale} is a global scale parameter and $\boldsymbol{\pi}_{elng}$ represents rest of the fit parameters.

The next step is standard forward kinematics, i.e., hierarchical composition of transformations which correspond to the rotations of individual joints (appearing as components of $\boldsymbol{\theta}$) and coordinate transformations between the individual joints. This is analogous to traditional forward kinematics models used in robotics Murray et al. [1994], with the only difference that in our model, the lengths of the bones can change according to the $\boldsymbol{\pi}$ parameters. If we denote the resulting transformation from the rest pose to the world space as $\text{FK}(\boldsymbol{\theta}, \boldsymbol{\pi})$, the entire rig function can be written as composition:

$$\text{Rig}(\boldsymbol{\theta}, \boldsymbol{\pi}) = \text{FK}(\boldsymbol{\theta}, \boldsymbol{\pi})\text{BoneFit}(\boldsymbol{\pi}), \tag{4.5}$$

where we assume the FK function returns a stack of homogeneous matrices which are applied to each of the rescaled rest pose bone samples returned by BoneFit.

Elastic potential E_{def}. Elastic models of biological soft tissues have received considerable attention both in the biomechanics Weiss et al. [1996], Fung [2013] as well as computer graphics communities Teran et al. [2005a,b], Sifakis et al. [2005], Lee et al. [2009], Sifakis and Barbic [2012]. Neohookean hyper-elastic materials have been found to function well in recent work Bickel et al. [2012],

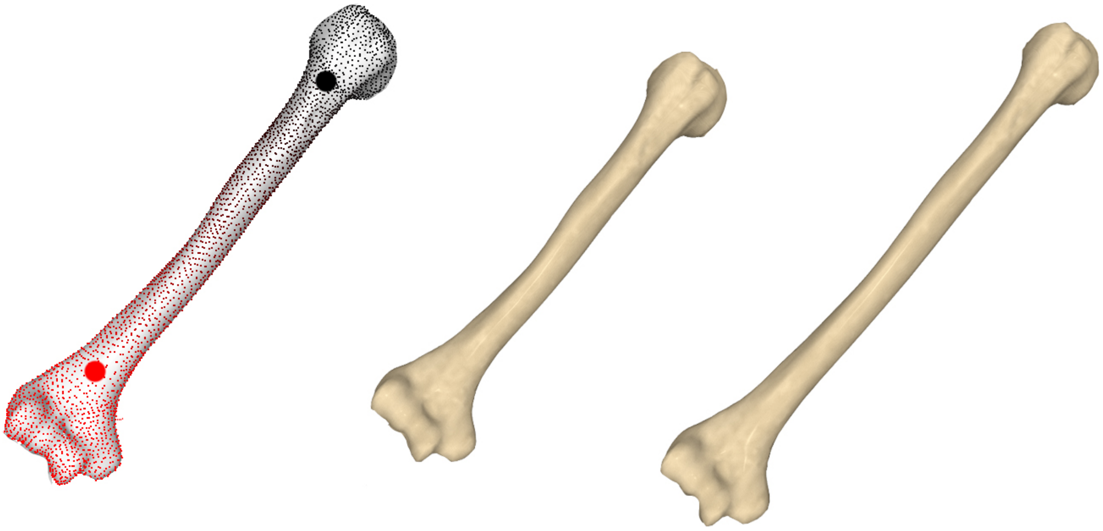


Figure 4.10: Example of the humerus bone elongation preserving shape of bone heads using two deformation handles and precomputed bounded biharmonic weights.

Skouras et al. [2014]. Their advantage is realistic modeling of large compression – when an element degenerates, the Neohookean energy approaches infinity, as such configuration is not physically realistic. However, for applications in computer graphics, this behavior can be problematic, because as shown by Irving et al. [Irving et al., 2004], inverted tetrahedra may be necessary to capture large deformations without resorting to remeshing. Increasing the mesh resolution can avoid these problems, but the resolution required to avoid all inversions would be prohibitively high; consider, e.g., the narrow space between cartilages of two bones connected by a joint. One possible solution is the popular corotated elastic model, which penalizes inverted elements by finite energies, i.e., allowing elements to invert if they are forced to do so. In the core of corotated elasticity is the following term: $\|\mathbf{D}_S\mathbf{D}_M^{-1} - \mathbf{R}\|_F^2$, where $\mathbf{D}_M$ and $\mathbf{D}_S$ are edge direction matrices in the material (i.e., reference) space and the deformed space (this notation is consistent with the tutorial of Sifakis and Barbic [Sifakis and Barbic, 2012]). The matrix $\mathbf{R} \in SO(3)$ is found by projecting $\mathbf{D}_S\mathbf{D}_M^{-1}$ onto the closest rotation.

Even though the classical corotated model is robust enough for use in a production environment McAdams et al. [2011b], it has a significant problem for our *inverse* problem, where we are optimizing also over the rest pose; i.e., in our setting, the matrices $\mathbf{D}_M$ are no longer constant. Unfortunately, we found that the inversion of the $\mathbf{D}_M$ matrices poses serious numerical problems when rest pose tetrahedra become close to degenerate; i.e., the $\mathbf{D}_M$ matrices become close to singular. This is problematic even if there is just a single degenerate tetrahedron present in the entire optimization.

To avoid these numerical difficulties, we use the following energy:

$$E_{\text{def}}(\mathbf{X}^{\text{src}}, \mathbf{X}) = \sum_i k_i \|\mathbf{D}_{S,i} - \mathbf{R}_i\mathbf{D}_{M,i}\|_F^2, \tag{4.6}$$

where the index i goes over all tets and $k_i \geq 0$ is stiffness of the i-th tet. Note that $\mathbf{D}_{M,i}$ depends linearly on $\mathbf{X}_i^{\text{src}}$, $\mathbf{D}_{S,i}$ depends linearly on $\mathbf{X}_i$ and $\mathbf{R}_i$ are rotation matrices minimizing the value of $E_{\text{def}}(\mathbf{X}^{\text{src}}, \mathbf{X})$. This optimal $\mathbf{R}_i$ can be computed by forming the signed SVD of $\mathbf{D}_{S,i}\mathbf{D}_{M,i}^{\mathsf{T}}$ and replacing the matrix of singular values with an identity matrix. We call this energy "symmetric as-rigid-as-possible" because $\|\mathbf{D}_{S,i} - \mathbf{R}_i\mathbf{D}_{M,i}\|_F = \|\mathbf{R}_i^{\mathsf{T}}\mathbf{D}_{S,i} - \mathbf{D}_{M,i}\|_F$, i.e., the rest pose and the deformed pose can be interchanged without changing the value of the energy. Perhaps more importantly, there is no need to invert the rest pose edge matrices $\mathbf{D}_{M,i}$, avoiding the numerical difficulties of the classical corotated model. Another advantage to the corotated model is that we do not need any volume weighting term such as $\frac{1}{6}|\det(\mathbf{D}_{M,i})|$ Sifakis and Barbic [2012], because our units do not cancel as in the $\mathbf{D}_S\mathbf{D}_M^{-1}$ term; i.e., larger tets automatically contribute more to the total energy than smaller ones. See the appendix for more details.

The *stiffness* k_i of each tetrahedron is computed as a weighted average of materials overlapping this tetrahedron. Note that even though our tet-mesh conforms to bone sample vertices, it does *not* conform to the full polygonal boundaries of the bones or muscles (which would require prohibitively high-resolution tet-meshes). Similarly to Lee et al. [Lee et al., 2009], we define the stiffness of each tetrahedron as $(\sum_t V_t k_t)/(\sum_t V_t)$, where t indexes individual material types (bones, tendons, muscles, generic soft tissues), $k_t > 0$ represents stiffness of each of the materials and V_t is the volume of a tetrahedron occupied by each component (bone, tendon, muscle, and generic soft tissues account for the remaining volume). We estimate V_t using Monte Carlo sampling (high accuracy is not necessary). See Section 4.9 for more details.

Handling Collisions We treat collisions in a fashion similar to McAdams et al. [McAdams et al., 2011b]. We detect tet-tet collisions using a fast bounding box sequence intersection algorithm Zomorodian and Edelsbrunner [2000]. For efficiency reasons, only selected regions near the joints are considered for collision processing, as these are the most common places where self-intersections occur. For example, our system does not try to detect or resolve pose-induced collisions such as hand touching the belly. The detected collisions are handled by instantiating temporary anisotropic springs that project the colliding vertices $\mathbf{X}$ out of the collision, to the surface of the tetrahedral mesh:

$$E_{\text{col}}(\mathbf{X}) = \left(\mathbf{n}_{\mathbf{\Pi}(\mathbf{X})}{}^{\mathsf{T}}(\mathbf{X} - \mathbf{\Pi}(\mathbf{X}))\right)^2, \tag{4.7}$$

where $\mathbf{\Pi}(\mathbf{X})$ is the projection of $\mathbf{X}$ onto the surface of the tetrahedral mesh, encoded by the barycenters of the closest surface triangle, and $\mathbf{n}_{\mathbf{\Pi}(\mathbf{X})}$ is the normal at the projected surface triangle. This anisotropy is helpful by allowing for sliding along the tangent plane at the projected surface point McAdams et al. [2011b]. The E_{col} energy potential is removed once the corresponding vertices are no longer in contact.

Muscle growth. Our symmetric as-rigid-as-possible (ARAP) elastic model can be extended to account for muscle growth Saito et al. [2015]. We accomplish this by replacing E_{def} with the following energy for the tetrahedra containing muscles:

$$E_{\text{muscle}}(\mathbf{X}^{\text{src}}, \mathbf{X}, \boldsymbol{\alpha}) = \sum_i \|\mathbf{D}_{S,i} - \mathbf{R}_i\mathbf{B}_i\mathbf{S}(\alpha_i)\mathbf{B}_i^{\mathsf{T}}\mathbf{D}_{M,i}\|_F^2, \tag{4.8}$$

which differs from the symmetric ARAP model by the term $\mathbf{B}_i\mathbf{S}(\alpha_i)\mathbf{B}_i^{\mathsf{T}}$ accounting for muscle growth. Specifically, the orthonormal matrix $\mathbf{B}_i$ is a change of coordinates which transforms the x-axis to align with the fiber directions of the i^{th} tetrahedron (Figure 4.4). The matrix $\mathbf{S}(\alpha_i)$ is a scaling matrix in the y and z-axes, which allows for simulating muscle shape changes due to atrophy or hypertrophy:

$$\mathbf{S}(\alpha_i) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha_i & 0 \\ 0 & 0 & \alpha_i \end{pmatrix}. \tag{4.9}$$

4.5 Inverse Problem

The input of our algorithm is a set of scans corresponding to various poses of a given human subject (see Figure 4.2). First, the input scans are registered against the skin surface mesh of our template body model $\mathbf{X}^{\text{tmpl}}$, i.e., deforming $\mathbf{X}^{\text{tmpl}}$ until it is in close correspondence with the target scans. We use a non-rigid ICP procedure Rusinkiewicz and Levoy [2001], explained in more detail in Section 4.5.1. We denote the resulting registered meshes as $\mathbf{T}_k$, where $k = 1 \ldots \text{numScans}$. The goal of inverse body modeling is to recover the subject-specific body shape in the rest pose $\mathbf{X}^{\text{pers}}$. Note that this configuration is devoid of the effects of gravity (as if in zero-gravity environment), because the gravity forces are added during the quasi-static solve in the forward skinning process (Eq. 4.1). In addition to determining $\mathbf{X}^{\text{pers}}$, we also have to solve for the bone fitting parameters $\boldsymbol{\pi}$ and joint angles $\boldsymbol{\theta}_k$, where k indexes individual poses, $k = 1 \ldots \text{numScans}$. The fitting parameters $\boldsymbol{\pi}$ are fixed for a given human being, but the joint angles $\boldsymbol{\theta}_k$ vary from pose to pose. We need to find the values of $\mathbf{X}^{\text{pers}}$, $\boldsymbol{\pi}$, and $\boldsymbol{\theta}_k$ such that the forward skinning function $\text{Skin}(\mathbf{X}^{\text{pers}}, \boldsymbol{\theta}_k, \boldsymbol{\pi})$ produces shapes as close as possible to $\mathbf{T}_k$. Because Skin is a complicated implicitly defined non-linear function, we introduce auxiliary variables $\mathbf{X}_k^{\text{arti}}$ for the personalized and articulated (posed) body shapes. When the inverse body modeling process is complete, we will have $\mathbf{X}_k^{\text{arti}} = \text{Skin}(\mathbf{X}^{\text{pers}}, \boldsymbol{\theta}_k, \boldsymbol{\pi})$; however, this equality does not have to hold in the intermediate steps of our optimization pipeline.

Targeting term. We formalize the requirement of $\mathbf{X}_k^{\text{arti}}$ aligning as closely as possible with $\mathbf{T}_k$ using the following "targeting term", which is the main objective of our optimization:

$$E_{\text{targ}}(\mathbf{X}_k^{\text{arti}}) = \sum_k \|\mathbf{N}_k^{\mathsf{T}}(\mathbf{S}^{\text{skin}}\mathbf{X}_k^{\text{arti}} - \mathbf{S}_k^{\text{corsp}}\mathbf{T}_k)\|^2, \tag{4.10}$$

where $\mathbf{N}_k$ is a matrix of stacked scan normals, $\mathbf{S}^{\text{skin}}$ is a binary selector matrix of surface vertices, and $\mathbf{S}_k^{\text{corsp}}$ is a matrix of barycentric coordinates that allows us to depart from the initial registration in order to account for imperfections in the initial correspondences. This is also why we use this "point-to-plane" objective which allows for sliding of the skin vertices of $\mathbf{X}_k^{\text{arti}}$ along their corresponding tangent planes at $\mathbf{T}_k$. The matrix $\mathbf{S}_k^{\text{corsp}}$ is initialized to the identity (i.e., trusting the initial registration as described in Section 4.5.1) and after each iteration of the optimization process, we search for new correspondences. Specifically, for every skin vertex of $\mathbf{X}_k^{\text{arti}}$, we search for closest point of $\mathbf{T}_k$, rejecting pairs further than 5 cm away or with normals differing by more than 30 degrees Rusinkiewicz and Levoy [2001].

Reconstruction. Inverse body modeling can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{X}^{\text{pers}},\mathbf{X}_k^{\text{arti}},\boldsymbol{\pi},\boldsymbol{\theta}_k} \quad & E_{\text{targ}}(\mathbf{X}_k^{\text{arti}}) + E_{\text{reg}}(\mathbf{X}^{\text{pers}},\boldsymbol{\pi}) \\ \text{subject to } & \nabla_{\mathbf{X}_k^{\text{arti}}} E_{\text{skin}}(\mathbf{X}^{\text{pers}},\mathbf{X}_k^{\text{arti}},\boldsymbol{\theta}_k,\boldsymbol{\pi}) = 0 \end{aligned} \tag{4.11}$$

where $k = 1 \dots \text{numScans}$ as before. The equality constraints require the posed shapes $\mathbf{X}_k^{\text{arti}}$ to be exactly in quasi-static equilibrium; however, these constraints will be relaxed during our numerical solution procedure described below.

But first, let us explain the regularization term $E_{\text{reg}}(\mathbf{X}^{\text{pers}},\boldsymbol{\pi})$. Reconstructing anatomical models from surface scans only is an ill-posed problem, because we lack direct measurements from the *inside* of the human body. Instead, we rely on anatomical priors to rule out unlikely or even unnatural anatomies. We use

$$\begin{aligned} E_{\text{reg}}(\mathbf{X}^{\text{pers}},\boldsymbol{\pi}) = \text{BoneFlesh}(\mathbf{X}^{\text{pers}},\boldsymbol{\theta}_0,\boldsymbol{\pi}) + E_{\text{def}}(\mathbf{X}^{\text{tmpl}},\mathbf{X}^{\text{pers}}) \\ + w_{\text{muscle}} E_{\text{muscle}}(\mathbf{X}^{\text{tmpl}},\mathbf{X}^{\text{pers}}). \end{aligned} \tag{4.12}$$

Even though the sum of the BoneFlesh and E_{def} terms is reminiscent of the forward skinning function, here these terms have a somewhat different function: they serve to explain deformations between individual human subjects, as opposed to poses of a single individual. The $\boldsymbol{\theta}_0$ vector of joint angles corresponds to the rest pose and the term $\text{BoneFlesh}(\mathbf{X}^{\text{pers}},\boldsymbol{\theta}_0,\boldsymbol{\pi})$ requires the personalized rest pose $\mathbf{X}^{\text{pers}}$ to align with the skeleton grown according to skeletal fitting parameters $\boldsymbol{\pi}$. The $E_{\text{def}}(\mathbf{X}^{\text{tmpl}},\mathbf{X}^{\text{pers}})$ term states that the deformation between $\mathbf{X}^{\text{tmpl}}$ and $\mathbf{X}^{\text{pers}}$ should be minimized. In other words, the personalized mesh needs to stretch or shrink according to the resized skeleton, but the shape should not depart too much from the initial template.

Finally, the $E_{\text{muscle}}(\mathbf{X}^{\text{tmpl}},\mathbf{X}^{\text{pers}})$ term penalizes shape changes which cannot be explained by muscle growth (the $\boldsymbol{\alpha}$ parameters are free). Our approach tries to explain as much shape variation as possible with biologically-inspired muscle growth. After that, fat growth is applied in the subcutaneous layer to match the surface shape of the target. The assumption is that muscle growth gives rise to different shapes than fat growth. The parameter $w_{\text{muscle}} \geq 0$ controls our confidence in this assumption and can be tuned by the user or based on external measurements; e.g., it could be obtained by the assessment of body fat percentage done by measuring the skin fold thickness, or standard body mass index (BMI) approximations. This worked well for both the overweight and muscular man (Fig. 4.14 third and first row). The effect of constrained muscle growth is also clearly visible in Fig. 4.19. When the muscle modeling can no longer explain target shape variations, the thin subcutaneous fat layer of tetrahedra (called “muscle envelope”) grows to match the target. Note that there is no gravitational potential acting on $\mathbf{X}^{\text{pers}}$; it only acts on the final articulated shapes $\mathbf{X}_k^{\text{arti}}$. In other words, our $\mathbf{X}^{\text{pers}}$ shape corresponds to the rest-pose body in a zero gravity environment Chen et al. [2014].

Penalty method. Equation 4.11 represents a non-convex constrained optimization problem that can be written in a general form as $\min f(\vec{\mathbf{x}})$ subject to $\vec{\mathbf{c}}(\vec{\mathbf{x}}) = 0$, where f is the objective and $\vec{\mathbf{c}}$ a vector function of constraints. We solve this optimization problem by converting it into a sequence of unconstrained optimization problems using the penalty method Nocedal and Wright [2006]. Each unconstrained subproblem has the following form: $\min f(\vec{\mathbf{x}}) + \gamma\|\vec{\mathbf{c}}(\vec{\mathbf{x}})\|^2$, where γ is the penalty weight. The γ parameter is progressively increased from 0 to γ_{max}.

Each γ-subproblem is solved using Newton's method with Hessian modification (Algorithm 3.2 in Nocedal and Wright [2006]). In particular, evaluating the exact Hessian matrix would be complicated because it contains third derivative terms (note that the constraints $\vec{\mathbf{c}}$ already contain first derivatives of the E_{skin} potential). Similarly to Bickel et al. [Bickel et al., 2012], we drop these third derivative terms. The approximate Hessian is further modified by adding scalar multiple of the identity matrix to ensure positive definiteness. Having determined the descent direction, we calculate appropriate step size using backtracking line search. We note that alternative numerical solution procedures are possible, e.g., the Augmented Lagrangian Method, however, we found that our quasi-Newton penalty method converges rapidly in our experiments.

4.5.1 Registration

In this section we describe our method to obtain the initial registration between our template model $\mathbf{X}^{\text{tmpl}}$ and the input scans $\tilde{\mathbf{T}}_1, \ldots, \tilde{\mathbf{T}}_{\text{numScans}}$, which are unstructured triangle meshes with noise, holes, or other imperfections. We use a non-rigid ICP procedure which deforms $\mathbf{X}^{\text{tmpl}}$ into $\mathbf{T}_1, \ldots, \mathbf{T}_{\text{numScans}}$ such that each $\mathbf{T}_k$ is well aligned with its corresponding scan $\tilde{\mathbf{T}}_k$. We initialize the process with approximately 15 landmark points, interactively selected by the user in our GUI. We use the tet-mesh associated with $\mathbf{X}^{\text{tmpl}}$ to define a regularization energy for non-rigid ICP. Specifically, we use our symmetric ARAP energy (Eq. 4.6) with uniform stiffness k_i for all tets. We do not even account for the rigidity of the bones, i.e., we treat the entire template tet-mesh as a jellyfish. This approximation is sufficient to establish good initial correspondences, which will be refined in subsequent iterations of our optimization process.

4.5.2 Reconstruction pipeline

Since the inverse problem formulated in Equation 4.11 is highly non-linear and the search space is large (e.g. up to 200k variables for our experiments), it can be challenging to find a correct solution and avoid numerical difficulties. Therefore, we split the optimization into several sub-problems to obtain well-defined initial values that approximate the solution. We found this step important to speedup the convergence and avoid local minima due to the non-linearity of some terms.

After registration, we start the reconstruction process by approximating skeleton parameters. We first optimize for global translation, rotation and scaling of the skeleton for each scan. Next, we enable the articulation parameters $\boldsymbol{\theta}$ to fit scan poses. After that, we optimize the rest of the skeleton variables including $\boldsymbol{\pi}$. This concludes the initial skeleton fitting and we move onto a next phase, which we call "forward initialization". In this phase we disable minimization over the $\mathbf{X}^{\text{pers}}$ variable in Eq. 4.11 to fit the scans without changing the original rest pose. Finally, we optimize over all variables (including $\mathbf{X}^{\text{pers}}$), taking advantage of good initial estimates computed in the previous steps. Our symmetric ARAP energy is particularly important to avoid numerical difficulties in evaluating the E_{def} and E_{muscle} terms.

4.6 Symmetric as-rigid-as-possible energy

Our reconstruction algorithm is formulated as an inverse problem which caused many issues in our experiments in terms of stability and numerical accuracy. At first, we implemented the optimization using standard ARAP elastic potential which proved robust for the forward simulation. However, for the rest pose optimization, we found that the inverse of the reference shape matrix D_M (which was no longer constant) caused problems during the reconstruction in cases of degeneration or inversion of tetrahedra.

To illustrate the problem, we formulate a simple 1D ARAP-like deformation energy:

$$E_{sdef}(l_P, l_R) = k \left\| \frac{l_P}{l_R} - 1 \right\|^2 \tag{4.13}$$

where k is the stiffness, l_R is a rest-pose and l_P is a deformed element length.

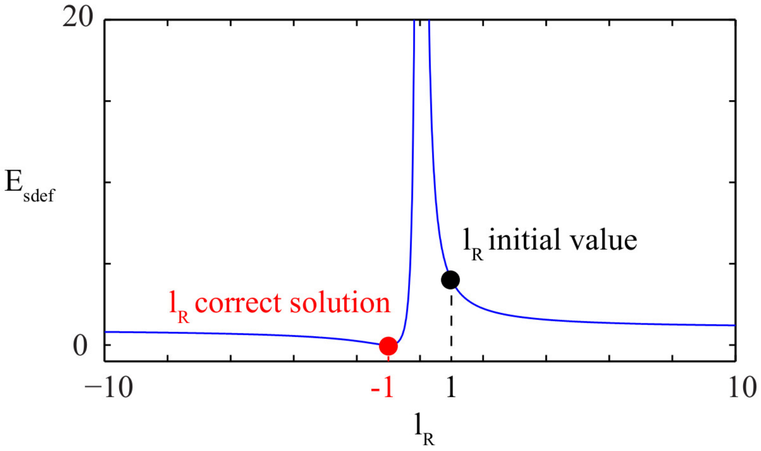


Figure 4.11: Illustration of the rest pose optimization instabilities of the ARAP-like energy in 1D caused by inverted element. Newton's method does not converge to the correct solution of the minimization problem $l_R = -1$ (red dot) when initialized as $l_R = 1$ (black dot).

The goal of the inverse problem is to find a rest-pose that minimizes the deformation energy $l_R = \arg\min_{l_R} E_{sdef}(l_P, l_R)$. The problem appears when the current configuration has element l_P inverted ($l_P = -1$) but undeformed in the rest-pose ($l_R = 1$). A correct solution is $l_R = -1$; however, the singularity of E_{sdef} at $l_R = 0$ causes Newton's method to iterate towards wrong result as shown in Fig. 4.11. Fig. 4.12 shows comparison of forward simulation of ARAP and symmetric ARAP energies.

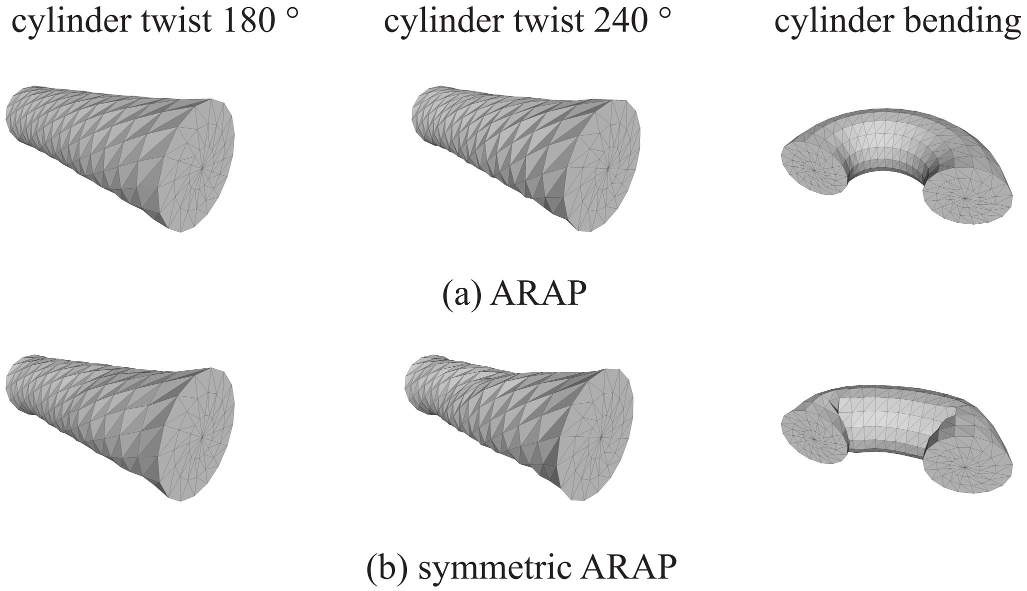


Figure 4.12: We compare results of the forward simulation of a cylinder deformation using the ARAP (a) and our symmetric ARAP energy (b). Although the symmetric ARAP energy converges to slightly less smooth results, it is much more robust in inverse body modeling. Neither simulation includes a volume preservation term which could be used to improve the visual quality of the results.

4.7 Animation

The resulting personalized body model $(\mathbf{X}^{\text{pers}}, \boldsymbol{\pi})$ is ready for physics-based animation. As input, we provide a time-varying sequence of joint angles $\boldsymbol{\theta}_j$, where the index j samples discrete time intervals (corresponding, e.g., to a constant time step such as $1/30s$). The animated joint angles can come from various sources such as keyframe animation or from retargeted motion capture data. This is particularly easy to achieve by using a subset of the functionality of our optimization framework.

But first, let us explain how to introduce dynamic effects, such as flesh jiggling. In our physics-based framework, this can be naturally achieved by switching from quasi-statics to full dynamics simulation. Assuming the widely used Implicit Euler time integration, this is as simple as adding an extra convex quadratic term to the energy terms in the $E_{\text{skin}}(\mathbf{X}^{\text{src}}, \mathbf{X}, \boldsymbol{\theta}_k, \boldsymbol{\pi})$ function (Eq. 4.2). This "inertial" term introduces history dependence, i.e., accounts for Newton's first law (which is ignored in quasi-statics). Specifically, let us denote the animated body shape as $\mathbf{X}_j^{\text{anim}}$, where j again indexes discrete time steps. We assume that $\mathbf{X}_0^{\text{anim}}$ and $\mathbf{X}_1^{\text{anim}}$ are provided as initial conditions (typically starting with zero velocities, i.e., $\mathbf{X}_0^{\text{anim}} = \mathbf{X}_1^{\text{anim}}$). The inertial term can be defined as:

$$E_{\text{inert}}(\mathbf{X}) = \frac{1}{2h^2}\|\mathbf{M}^{1/2}(\mathbf{X} - 2\mathbf{X}_j^{\text{anim}} + \mathbf{X}_{j-1}^{\text{anim}})\|^2 \tag{4.14}$$

where $\mathbf{M}$ is a diagonal mass matrix and h is the time step. This term can be derived from the Implicit Euler integration rules, which can be found e.g. in Bouaziz et al. [2014].

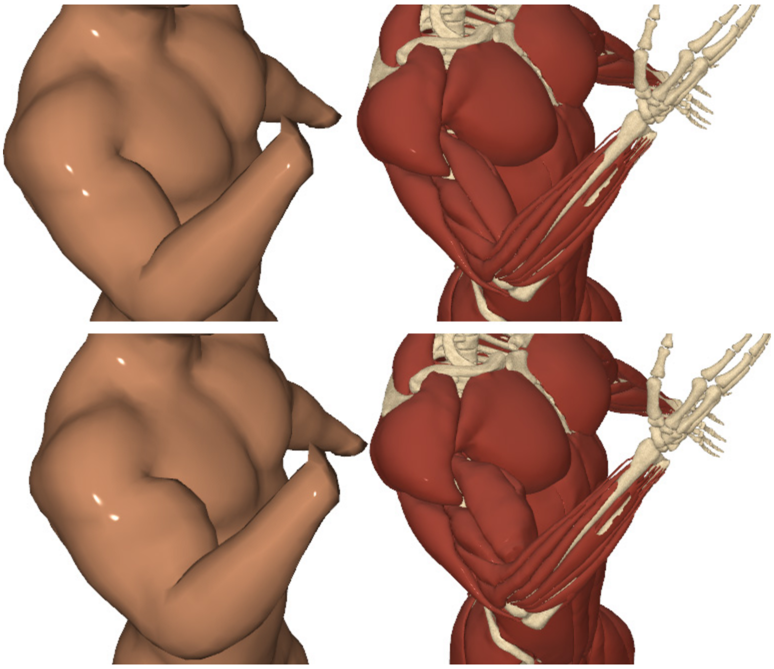


Figure 4.13: Our physics-based animation approach allows for animating pose-specific muscle shape changes due to muscle contractions. The upper images show the shape of the arm and muscles in a flexing pose, and the lower images show the effect of contracting the biceps muscle in the same pose.

In addition to the inertial term, we also add the collision avoidance potential E_{col} discussed in Section 4.4. Gravity potential is also accounted for as described already in Eq. 4.2.

The physics-based animation framework is quite versatile and in addition to supporting the effects of inertia, collisions, and gravity, we can also add muscle contraction forces. We assume that time-varying muscle activation signals are provided by the user. These can be e.g., keyframed, which is common in professional VFX animation systems WETA digital [2013], or calculated using inverse dynamics models Lee et al. [2009]. Let us denote the muscle activation signals as $\boldsymbol{\beta}_j$, where j indexes discrete time steps as before. The muscle contraction potential is similar to the muscle growth potential (Eq. 4.8), however, instead of the rest-pose growth matrix $S(\alpha_i)$ for each tetrahedron i (Eq. 4.9) we use the following matrix:

$$\mathbf{S}(\beta_{i,j}) = \begin{pmatrix} \beta_{i,j}^{-1} & 0 & 0 \\ 0 & \sqrt{\beta_{i,j}} & 0 \\ 0 & 0 & \sqrt{\beta_{i,j}} \end{pmatrix} \tag{4.15}$$

which accounts for the volume preserving nature of muscle contraction due to high water content in soft biological tissues Weiss et al. [1996]. Mathematically, this is modeled by the fact that the determinant of matrix $\mathbf{S}(\beta_{i,j})$ is one, resulting in the characteristic bulging behavior of contracting muscles (see Figure 4.13 for an example). Note that the muscle growth scaling matrix $\mathbf{S}(\alpha_i)$ (Eq. 4.9) does not have determinant one because it accounts for *growth*, which is of course not volume conserving.

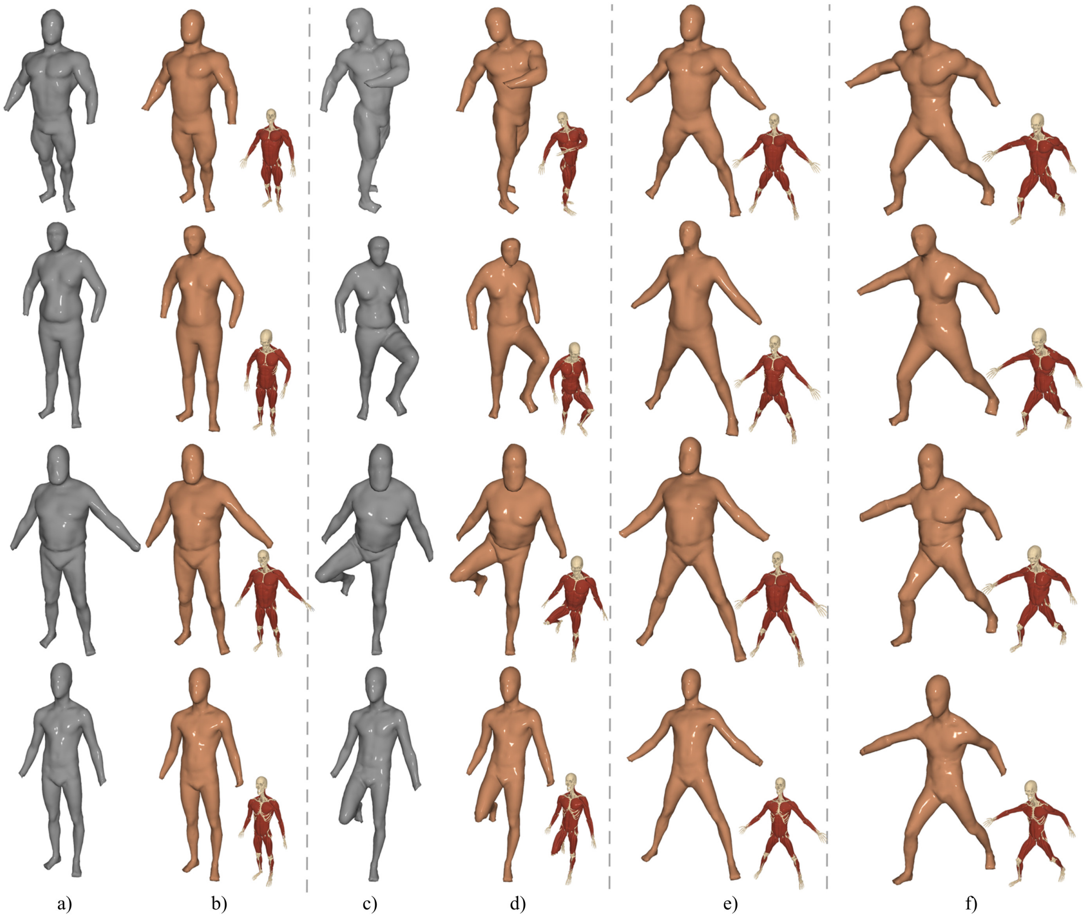


Figure 4.14: Registered 3D surface scans of our test subjects in two different poses (a, c) and corresponding reconstructions using our anatomical physics-based model (b, d). Note that the shapes are quite similar. We also show our optimized rest pose $\mathbf{X}^{\text{pers}}$ (e) and a novel, unseen pose synthesized using our forward skinning model (f).

4.8 Results

We performed our experiments on 3D surface scans with diverse quality and resolution. Specifically, we tested our reconstructions on publicly available good quality 3D surface scans obtained from the FAUST dataset Bogo et al. [2014] and database of Hasler et al. [Hasler et al., 2009], and on high quality commercially available scans TEN 24 [2016]. Additionally, we also experimented with low resolution scans captured using the Microsoft Kinect with the Skanect Pro registration software.

Reconstruction accuracy. We have successfully reconstructed targets with various body types and skeletal variations including a muscular bodybuilder, subjects with apparent subcutaneous fat, as well as a slim actor, see Figure 4.14. We used between 2 to 5 scans for each subject depending on the quality of scans and diversity of the poses. Although it would be possible to use only a single scan in our method (similarly to Dicko et al. [Dicko et al., 2013]), this would mean the underlying anatomical model would be less well determined. In particular,

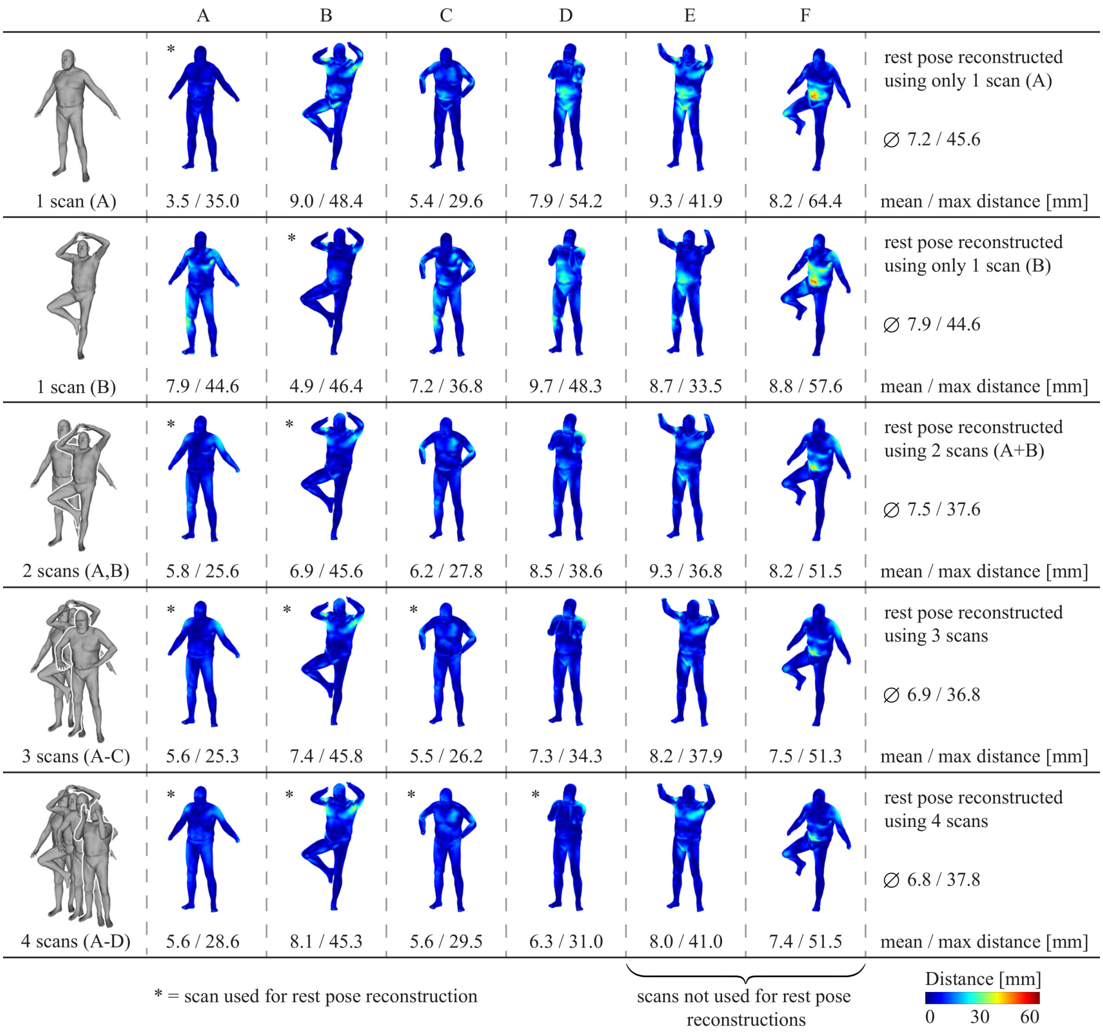


Figure 4.15: Evaluation experiment showing the effects of using different input scans (rows 1 and 2), as well as varying the number of input scans (rows 3, 4, and 5) on the rest pose reconstruction process. The mean and maximum point-to-plane distances between the input scans and our fits (columns 2, 3, 4, 5 marked with *), as well as between our posed reconstr. and not-seen-before scans (columns 6 and 7) are shown.

we observed ambiguities when optimizing for subject specific variations in bone lengths. For example, given one 3D surface scan with the actor with straight limbs, it is very difficult to accurately determine the locations of the joints. Jointly optimizing over scans of multiple poses, e.g. adding a scan with bent limbs, helps to eliminate this uncertainty, as the optimization algorithm places the joint in the most appropriate location. In Figure 4.14 we demonstrate the accuracy of our approach in terms of matching the input 3D scans. Our results show that our physics-based model can reproduce high quality body shapes with a close visual similarity to the scans. Moreover, in Figure 4.15 we show how the fitting accuracy improves when increasing the number of input scans.

Gravitational effects. Another advantage of using multiple scans is that it greatly reduces the ambiguity due to gravitational effects and self-collisions of the

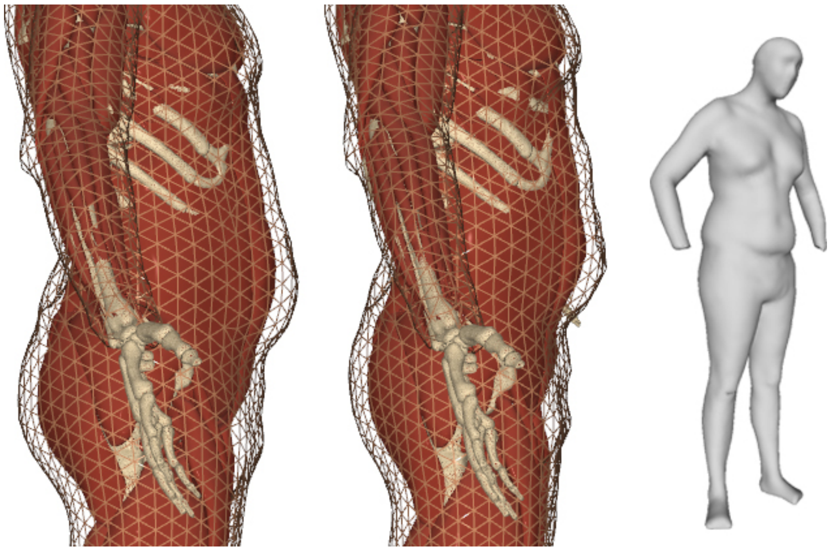


Figure 4.16: Example of the effect of gravity on the rest pose reconstruction process. The figure on the left shows the result of the reconstruction without taking gravity into account. In the middle, gravity is taken into account and eliminated from the rest pose – note that the belly “floating” as if the body was submersed in water. This “zero gravity” rest pose matches the input scan (right) closely because gravity is added during the forward simulation process.

skin. In Figure 4.16 we show the effect of taking gravity into consideration during our inverse body modeling process. We aim to reconstruct the rest pose in zero gravity, because gravity will be added in the forward simulation process. Note that this is a challenging problem in its own right Chen et al. [2014].

Collisions. An example of collision handling during the forward animation phase is shown in Figure 4.17. Equally important is collision handling during inverse body modeling. When the input 3D scan contains body parts in contact, it means the measured shape was influenced by action-reaction forces preventing the flesh from inter-penetrating. Our E_{col} term estimates these contact forces and compensates for them during our inverse body modeling process. This results in recovering more accurate rest poses, as shown in Figure 4.18.

Comparison to Anatomy Transfer. Our approach has several key advantages over Anatomy Transfer Dicko et al. [2013] and its extensions Zhu et al. [2015]. First, our approach can take advantage of multiple scans in different poses, which leads to high reconstruction accuracy, as discussed above and shown in Figure 4.15. Second, Anatomy Transfer as well as its extensions Zhu et al. [2015] use only an approximate deformation model of biological soft tissues. In our method, we use more realistic growth models for the bones and muscles, which allows us to estimate the underlying anatomy more accurately. Specifically, our material-aware deformation can reconstruct subcutaneous fat and the shape of skeletal muscles more realistically as shown in Figure 4.19. Furthermore, our reconstruction process is fully automatic, without needing artistic input. This is particularly important for animations including inertial effects and secondary motions of soft tissues. Our method is also able to reconstruct the whole skeleton while preserving

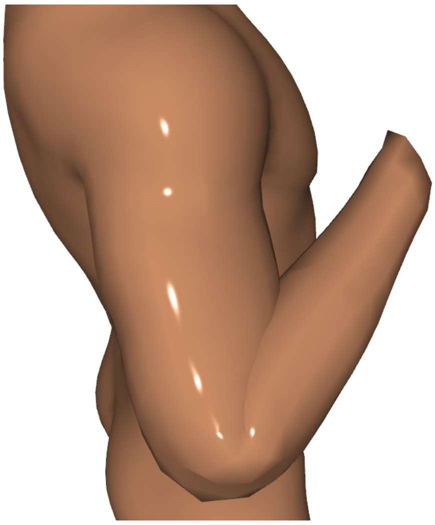


Figure 4.17: Forward simulation collision handling example.

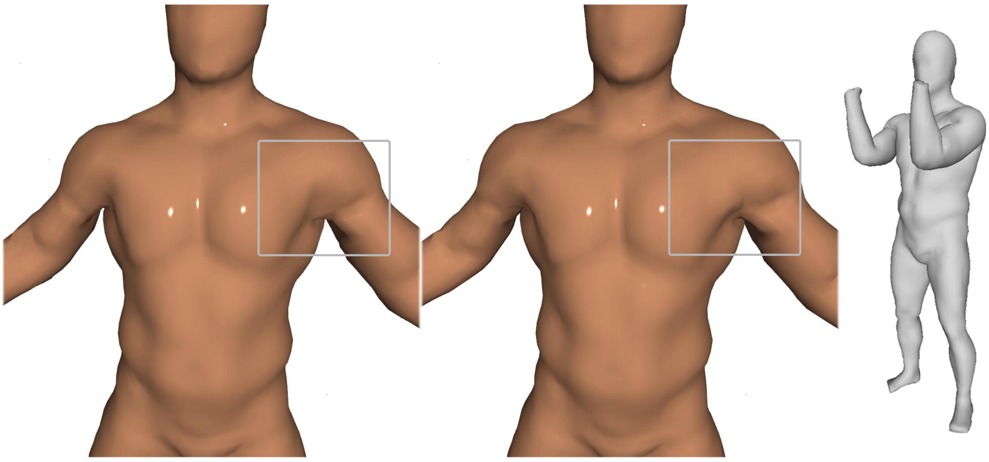


Figure 4.18: Example of collision handling during inverse body modeling. In this example, a single scan was used (shown in gray) in which the actor was pressing his arms against his body. Notice that the rest pose reconstruction on the left has the shape of the arm imprinted on the chest; the reconstruction on the right does take the collision forces into account and reaches a more realistic rest shape.

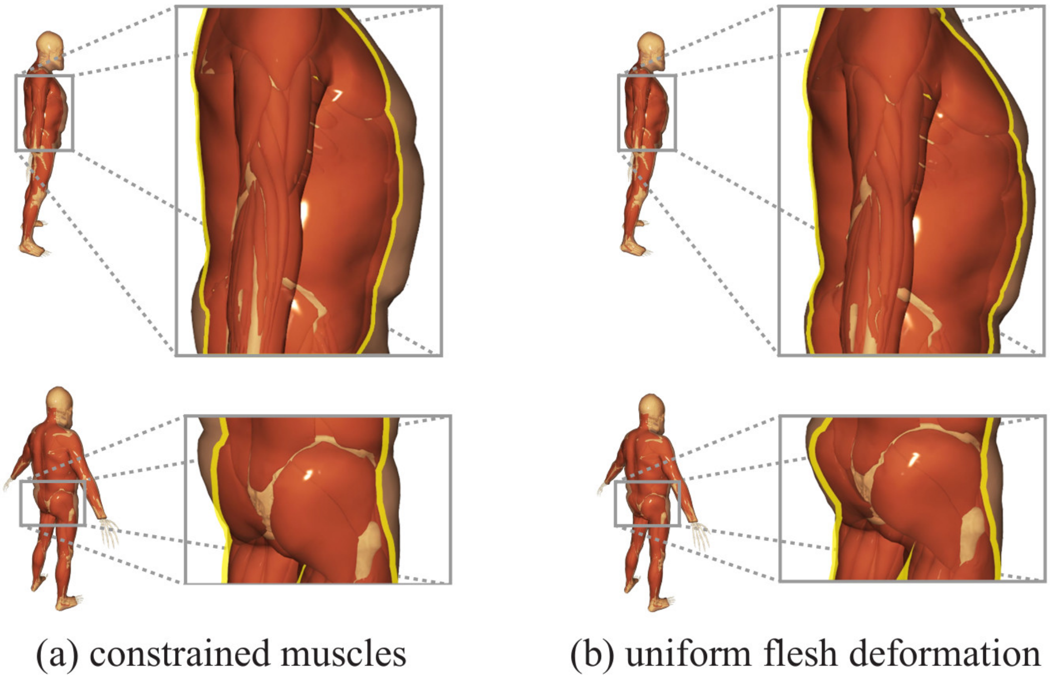


Figure 4.19: Example of material-aware deformations during inverse body modeling. In this comparison, muscle modeling is constrained by muscle fiber directions (a). The yellow outline shows that muscles are modeled more accurately and do not extend into regions that clearly should be fat in contrast to simple uniform flesh deformation used in (b). Material modeling is important for realistic simulation of inertial effects of soft tissues, such as subcutaneous fat.

realistic bone shapes, while Anatomy Transfer deforms skeleton using general affine transformations which results in non-realistic bone deformations.

Please also refer to the accompanying video A.1 to see more detailed results.

4.9 Implementation details

The geometric search data structures and algorithms used for the scan registration and collision detection are based on CGAL The CGAL Project [2016] and nanoflann Muja and Lowe [2014]. Numerical linear algebra is implemented using Eigen Guennebaud et al. [2010]. We benchmarked the performance on a consumer laptop with a 3.1 GHz Intel Core i7 processor and 32GB of main memory. For a complete rest pose optimization using 4 scans, we needed a total number of about 15 Newton iterations until convergence, with about 120s of computation time per iteration. The template model used for the results presented in this work has 12977 vertices, out of which 4901 are surface vertices, and 64164 tetrahedra. The skeleton used for rigging has 67 joints with a total of 52 articulation and 38 sizing parameters. There are 111 muscles in the template model.

In order to compute the contribution of each material to each body tetrahedron, we use a Monte Carlo sampling approach. For each muscle/tendon/bone tetrahedron T_m, we generate one sample for each mm^3 of the volume of T_m. Specifically, we generate random samples using a uniform distribution around the centroid of T_m until the desired number of samples is reached. Using those locations, we perform look-ups in the AABB tree of the body tetrahedrons T_b and count the contributions of those samples inside the body. In the forward simulation for the animation stage, we use a time step $h = 1/30s$, and we build the mass matrix $\mathbf{M}$ assuming uniform density of the material in the body, meaning that the per-vertex mass is proportional to the sum of the volumes of the tetrahedra in which that vertex is present. Our approach proved robust, and excessive parameter tuning was not needed. The material parameters we used to generate results are: $k_{\text{bone}} = 10^{-1}$, $k_{\text{def\_bone}} = 7 \times 10^{-4}$, $k_{\text{def\_tendon}} = 3 \times 10^{-4}$, $k_{\text{def\_muscle}} = 2 \times 10^{-4}$, $k_{\text{def\_soft\_tissue}} = 10^{-4}$, $k_{\text{muscle}} = 10^{-3}$. The penalty term γ was increased from 0 to $\gamma_{max} = 10^7$ by factors of 10. In our experiments, increasing γ further did not produce any visible differences.

4.10 Limitations and future work

We focus on capturing the physics of large- and medium-scale anatomical details, but we do not reconstruct hands or toes. We believe that these are research topics on their own which require specialized approaches. However, such techniques already exist and could be integrated in our body modeling framework.

In the visualizations of our experiments we noticed that the bones sometimes protrude through the muscles, which is most visible in the chest region. This is due to the soft non-conformal embedding of the bones in the tetrahedral mesh of the body, as well as due to the multi-material property of each body tetrahedron. These problems could be alleviated by increasing the resolution of the template model, which may lead to the necessity of applying more memory-efficient and performant optimization techniques. Another solution is to use hard constraints for the BoneFlesh term which would require a different class of algorithms to solve the constrained optimization problem, e.g. Augmented Lagrangian Method as discussed in Section 4.5.

We do not consider muscle shape changes in the posed scans, assuming all the muscles are in a relaxed stage or that they are not contracted significantly. While this holds true for most of the scans we used in our experiment, one can think of poses and situations in which correctly capturing the shape variation of muscles due to contractions becomes important. For example, using a scan of the bodybuilder flexing his arm muscles together with scans in which he was relaxed created issues in our optimization. However, once reconstructed, our anatomical models allow for simulating muscle contraction in the forward animation stage. A venue of future research would be to automatically extract muscle activations given the pose of the subject, and to normalize the shape changes due to contractions in the rest pose reconstruction problem.

The scans used in our experiments are static poses, in which the actor was in equilibrium. The reconstruction problem becomes much more complex when dynamics is added to the scans, e.g., by capturing a continuous stream of point clouds from an actor's performance. Our algorithms do not make any male-specific assumptions (we even removed the genitals) and should therefore work for females. The only issue is preparation of a template female body which would require non-trivial 3D modeling efforts. Stylized or imaginary characters such as Incredible Hulk are beyond the scope of this project, because we cannot capture their ground truth scans and also, their musculoskeletal structure may differ from human anatomy. This opens opportunities for another research project with emphasis on user interaction and anatomical modeling.

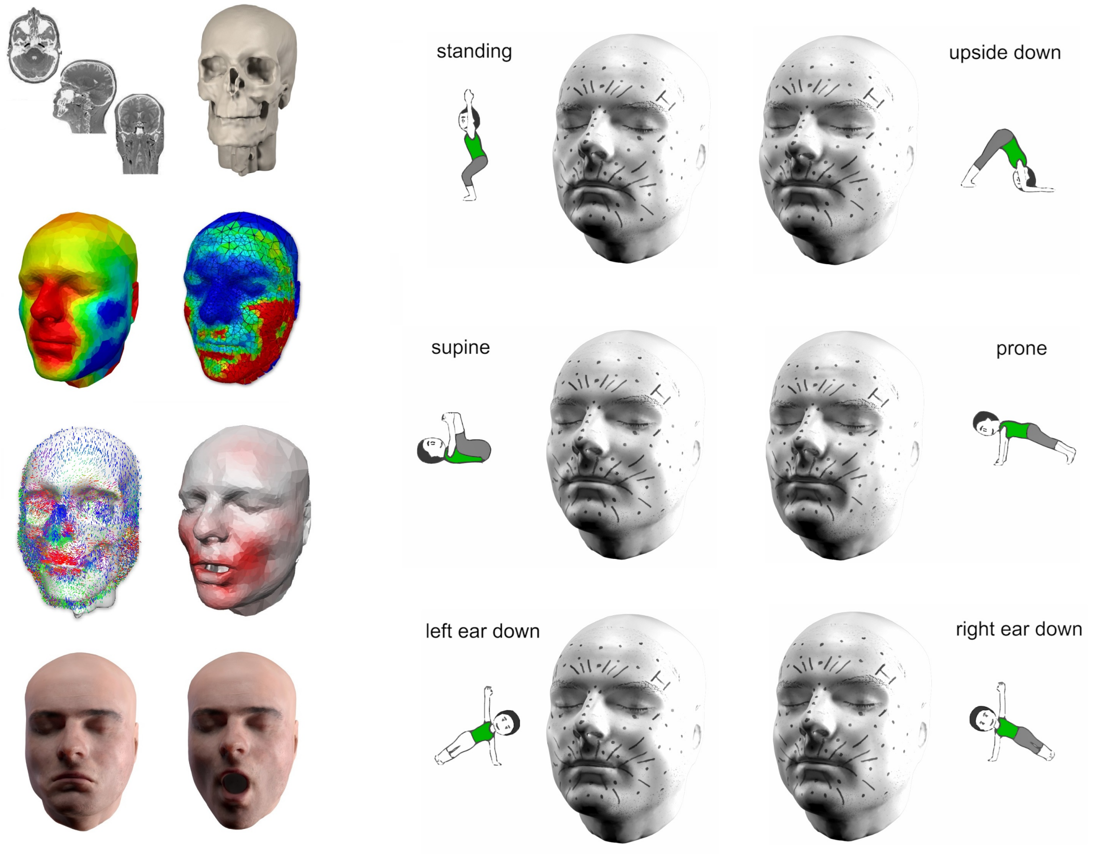


Figure 5.1: We build a static anatomical face model from the MRI and use 3D surface scans as training data to learn mechanical parameters that explain deformations of the real face using physics-based simulation.

5. Inverse Human Face Modeling

This chapter is based on two publications: [Kadleček and Kavan, 2019, Ichim et al., 2017]. In the first publication, all scientific contributions belong to the author and his supervisor Ladislav Kavan. The work is covered in Section 5.4 *Learning mechanical properties and muscle control.* The author implemented most of the code required for this project.

In the second co-authored publication, the author contributed by implementation of elastic models suited for both forward and inverse problems, skull and jaw kinematics, dynamics, plasticity and some applications of the reconstructed model including body mass index changes, inertial and wind simulations. This work is covered in Section 5.3 *Modeling and Animation with blendshape-type control.*

5.1 Introduction

Facial appearance plays an important role in many areas of life, including communication and interpersonal relationships. The human face is a fascinating mechanical system able to generate facial expressions by contractions of delicate facial muscles. Forces due to contractions of these muscles are transmitted to adjacent muscles, the bones and the skin via a network of connective tissues. Despite decades of progress, accurate biomechanical face modeling remains a challenge. Computer graphics typically employs direct deformation models such as blendshapes Parke [1972] which capture only the surface (skin) shapes and do not attempt to model the internal biomechanics or anatomy of the face.

Recent work [Ichim et al., 2016, Barrielle et al., 2016, Cong et al., 2016, Kozlov et al., 2017, Bao et al., 2018] proposes to avoid these shortcomings by augmenting the generative approach of blendshape animation with a simulation-based solution. One of their advantages is the ability to add realistic physics-based effects such as collisions or inertia. Physics-based simulation in computer graphics typically involves tuning of geometric and material properties to achieve the desired visual effect. This process is non-trivial and time consuming even for experienced technical artists. Measurement-based approaches for determining mechanical parameters of biological soft tissues have been developed in biomechanics, typically involving mechanical loading experiments with excised samples of tissues [Lapeer et al., 2011] or using a specifically designed aspiration device [Luboz et al., 2014].

One major difficulty in simulation-based approaches is to achieve the required level of realism, which is particularly challenging for facial animation, due to the heightened human sensitivity for facial motion perception [Bruce and Young, 1986]. Accurate simulation requires building a detailed volumetric face model that faithfully represents the shape and dynamics of the captured subject.

In the first part of this chapter, we approach this problem by combining easy-to-obtain facial surface scans with a template model that integrates rigid bone structures, active muscle tissues and passive flesh, fat, and skin layers in a fully volumetric simulation model of the human face (see Figure 5.4). Similarly to the approach we took in Section 4 for body reconstruction, we obtain a representation of the geometry and expression dynamics of the acquired person's face by scanning the subject in multiple facial poses. We then solve an inverse problem to estimate the activation parameters of the registered template rest pose in order to best reproduce the scanned expressions under activation. Subsequently, we can create new animations driven by muscle activations using a forward physics simulation that incorporates collision handling, volume preservation, inertia, and external forces such as wind forces or gravity. Muscle activations can be computed from a temporal sequence of blendshape weights, which enables straightforward integration into existing animation environments.

In the second part of this chapter, we chose another approach and built our own static model from MRI (see Figure 5.1 and Figure 5.13) which is typically very time consuming and still can not guarantee that the shapes of different anatomical structures are accurate due to low resolution and noise. In our case, we needed a better guess than the deformed template model, but we also did not want to spend too much time on processing MRI scans by segmenting separate muscles. For this reason, we only segmented (mostly automatically) clearly visible structures and proceeded with rough segmentation of the rest, leaving the discovery of anatomical details to our learning algorithm.

5.1.1 Contributions.

The main technical contributions of our work are:

- an inverse modeling adapting the anatomical face model to a series of 3D surface scans of a specific person,
- a computation of mechanical parameters of soft tissues from in-vivo surface measurements (3D scans),
- a flexible muscle activation model compatible with standard blendshape-driven facial animation,
- a more restrictive subject-specific fiber-based muscle activation model forcing more accurate passive tissue parameter estimation,
- a physics-based simulation method that retains realism even with significant external forces or substantial modifications of the face geometry and tissue material properties.

An important feature of physics-based approaches is that their parameters can be controlled to achieve the desired effects. In our case, the parameters include the stiffness of simulation elements, their rest shape volume, the static bone structure, or the muscle activation parameters. This detailed control facilitates numerous new applications that are difficult to achieve with existing methods. Examples we show in this chapter include

- slimming and fattening of the face by adapting the volume of soft tissue,
- simulation of corrective facial surgery, such as orthognathic surgery to correct for jaw malformations,
- dynamic interaction with external forces (e.g. wind) and objects (e.g. VR headsets),
- artistic editing of facial expression dynamics by modifying tissue stiffness or muscle behavior.

5.2 Related work

Facial animation dates back to the pioneering work Terzopoulos and Waters [1990], Blanz and Vetter [1999]. More recently, high-fidelity capture setups of actor's faces have become important in the film and game industries, such as Medusa system Beeler et al. [2010, 2011], Beeler and Bradley [2014] or other Alexander et al. [2010], von der Pahlen et al. [2014], Smith et al. [2017]. Method for accurate registration and stabilization have been explored, including kinematics models of a jaw Beeler and Bradley [2014], Yang et al. [2018], Zoss et al. [2018]. Industrial pipelines typically involve data-driven blendshape-type models which continue to be an important topic of research Lewis et al. [2014], Ichim et al. [2015], Zell et al. [2017], Li et al. [2017], Yoon et al. [2017]. In general, the chief advantage of blendshape-type approaches is direct control of skin deformations which translates into high visual fidelity. Achieving high quality with biomechanical facial modeling is harder, because facial muscles influence the shape of the skin indirectly and are sensitive to the precise geometric and material modeling of anatomical shapes Sifakis et al. [2005], Cong et al. [2015, 2016]. Despite these challenges, the quest for high-fidelity anatomical face models continues both in the academia and the industry Lan et al. [2017].

Physics-based anatomical models have been explored in computer graphics especially in the context of full-body animation Teran et al. [2003, 2005a,b], Lee et al. [2009]. Recent methods explore new numerical simulation strategies such as Eulerian-on-Lagrangian simulation Fan et al. [2013, 2014], Projective Dynamics Saito et al. [2015], Kadlecek et al. [2016], or a combination of data-driven and physics-based animation Kim et al. [2017]. While sharing the same biological underpinnings, the shape and function of facial muscles are quite different from large skeletal muscles such as the biceps Blemker et al. [2005]. In particular, facial muscles are very thin and are attached via connective tissues to each other and the skin; the primary function of facial muscles is generation of facial expressions Ekman and Friesen [1977].

Building upon seminal work Terzopoulos and Waters [1990], in recent years there has been a resurgence of interest in physics-based facial animation in computer graphics. Blendshape animation has been enriched with dynamics effects using mass-spring systems Ma et al. [2012], "Projective Dynamics"-based surface simulation Barrielle et al. [2016], or finite element simulation of an outer volumetric layer attached to an inner blendshape model Kozlov et al. [2017] (a hybrid approach effective also for full-body animation Kim et al. [2017]). Anatomical modeling is a non-trivial task which can be facilitated using automatic methods Stavness et al. [2014], which is especially challenging for facial modeling of stylized characters such as Yoda Cong et al. [2015].

Accurate control of facial expressions via simulated facial muscles is a hard problem. Cong and colleagues introduced "art-directed muscles" that allowed artists to sculpt facial muscle activations Cong et al. [2016]. This was further improved with careful modeling of facial anatomy Lan et al. [2017]. A different approach was explored by Ichim et al. [2016], who proposed "volumetric blendshapes", allowing all tetrahderons in a finite element simulation to activate and thus achieve desired shapes while enjoying the benefits of physics-based simulation. A blendshape-driven muscle control approach was also successfuly applied for fitting monocular RGB images with a fully differentiable pipeline Bao et al. [2018]. Recently, Ichim et al. [2017] fit a generalized muscle model to training data (3D scans of an actor in different facial expressions). While related to our approach, the muscle model of Ichim et al. [2017] can produce biologically unrealistic activations which compensate for the simplifying assumptions in material modeling. Specifically, the elasticity model used in Ichim et al. [2017] relies on a homogenous isotropic material with zero prestrain, which is only a crude approximation of the real mechanical behavior of facial soft tissues. In this work, we propose methods to learn heterogeneous, anisotropic materials with prestrain from data, allowing our model to predict realistic mechanical effects, such as changing directions of gravity.

Methods for learning material parameters from data have a long history in graphics Pai et al. [2001]. Bickel et al. [2009] captured non-linear heterogenous soft tissues by probing a deformable object and measuring its elastic response using computer vision techniques. Even though their results include a face example, the face model is passive (not actuated) and does not model muscles or prestrains. Robotic actuation has been applied to create animatronic facial replicas with the aid of 3D printing Bickel et al. [2012]. Despite some similarities, the mechanics of man-made robotic systems is quite different from living tissues. 3D printing of deformable objects that assume the desired shapes under gravity motivated development of novel numerical methods Chen et al. [2014], later extended with data-driven modeling of dynamics Wang et al. [2015].

Mechanical modeling of biological soft tissues has been extensively studied in biomechanics Weiss et al. [1996] and remains an active area of research aided with simulation platforms such as the open source ArtiSynth Lloyd et al. [2012], SOFA Faure et al. [2012], and FEBio Maas et al. [2012] or the commercial Abaqus Hibbitt et al. [2001]. In-vivo material parameter identification of the breast was studied from gravity loading data and plate compression Han et al. [2011]. Synthetic experiments showed that three gravity loading orientations were required to identify heterogeneous parameters of a silicon gel beam Gamage et al. [2011]. Barbarino et al. [2009] modeled elasticity of the face using an advanced Rubin-Bodner material Rubin and Bodner [2002], but did not consider muscle activations or formation of facial expressions. In a similar vein as Sifakis et al. [2005], an MRI-based biomechanical face model was constructed by Wu [Wu, 2013]. In contrast to this work, Wu [2013] finds muscle activations for target expressions manually and relies on accurate muscle segmentations from the MRI and material (Mooney-Rivlin) parameters from the literature Tran et al. [2007], Nazari et al. [2010].

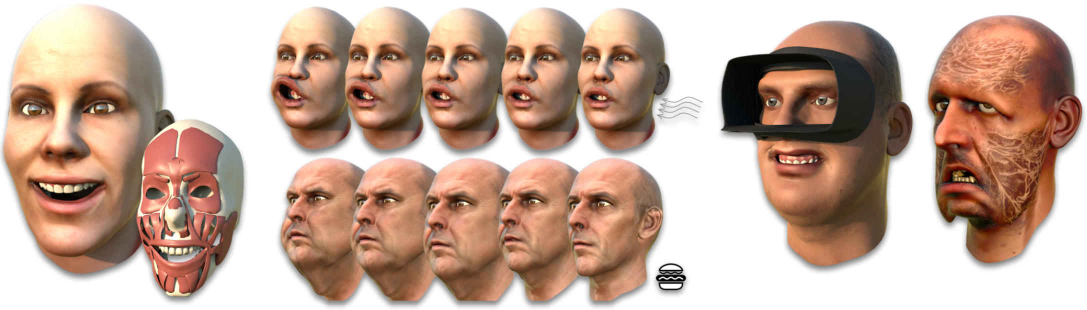


Figure 5.2: Physics-based simulation facilitates a number of advanced effects for facial animation, such as applying wind forces, fattening and slimming of the face, wearing a VR headset, and even turning into a zombie.

5.3 Modeling and Animation with blendshape-type control

In this section, we leverage a face template model that combines volumetric and surface elements as shown in Figure 5.4. Physics-based optimization is performed on a tetrahedralized volumetric model composed of rigid bones and deformable tissue. The latter is further separated into *active* muscles, and *passive* flesh and skin. Muscles actively deform to drive the dynamic motion of the face model. In order to control the animation, we augment the volumetric template with a surface blendshape basis that represents the facial expression space. This also provides an interface to the surface scans used to build actor-specific simulation models.

The core algorithmic components of our method are the inverse and forward physics simulation modules. Inverse physics is used in a model building stage to create a simulation-ready anatomical face model of a specific person. As input to this preprocessing stage, we assume a set of surface scans that are first transformed to a user-specific blendshape model. An anatomy transfer step warps the volumetric template towards the neutral expression of the blendshape model. Subsequently, our inverse physics solver computes suitable muscle activations of the simulation model to best approximate each expression blendshape.

Given the person-specific simulation model and corresponding muscle activation patterns, we can apply forward physics simulation to compute dynamic face articulations. This animation stage takes as input a temporal series of blendshape weights that are mapped to per-frame muscle activations. External effects such as gravity or object collisions can be incorporated in the simulation to support a wide range of dynamic effects (see Figure 5.2).

Figure 5.3 provides a visual summary of our physics-based face modeling and animation approach.

5.3.1 Template Face Model

Our approach starts from a *generic face model* – an anatomical face template corresponding to an average human subject (see Figure 5.4). We created this model from a commercially available anatomical data set Zygote [2016] that contains polygonal representations of the bones (the skull, the jaw, including

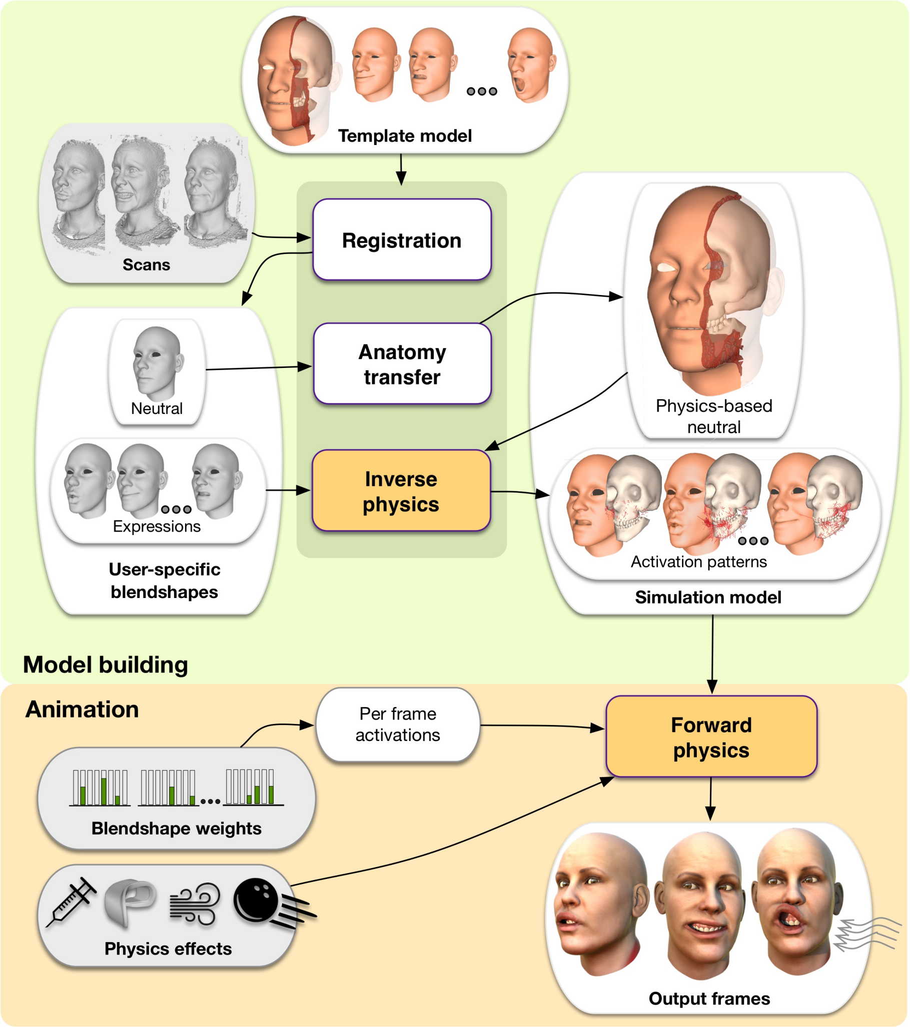


Figure 5.3: Schematic workflow of our method.

teeth), skin (including a realistic model of the oral cavity), and 33 facial muscles. Using the winding-number method of Jacobson et al. [Jacobson et al., 2013] we generate a tetrahedral mesh discretizing the soft tissue of the face. Our tet-mesh conforms to the skin and the bones, but not to the muscles, because a conforming discretization of the numerous thin facial muscles would require prohibitively many elements. Instead, we use non-conforming discretization where every tetrahedron can represent multiple types of soft tissues. We distinguish between two types of soft tissues: *active* corresponds to muscles, while *passive* corresponds to subcutaneous fat, connective tissue and the skin, i.e., tissue that is not voluntarily activated by neural signals (Figure 5.4-b).

Up to the accuracy of the discretization, the active layer corresponds to the union of all facial muscles, while the passive layer forms the region between the active layer and the skin and fills in areas between the bones. Even though this model is not as accurate as modeling every muscle individually, it captures the key fact that the shape of the skin is affected by facial muscles only *indirectly*, i.e., the contracted muscles deform passive soft tissue, which consequently induces skin deformations.

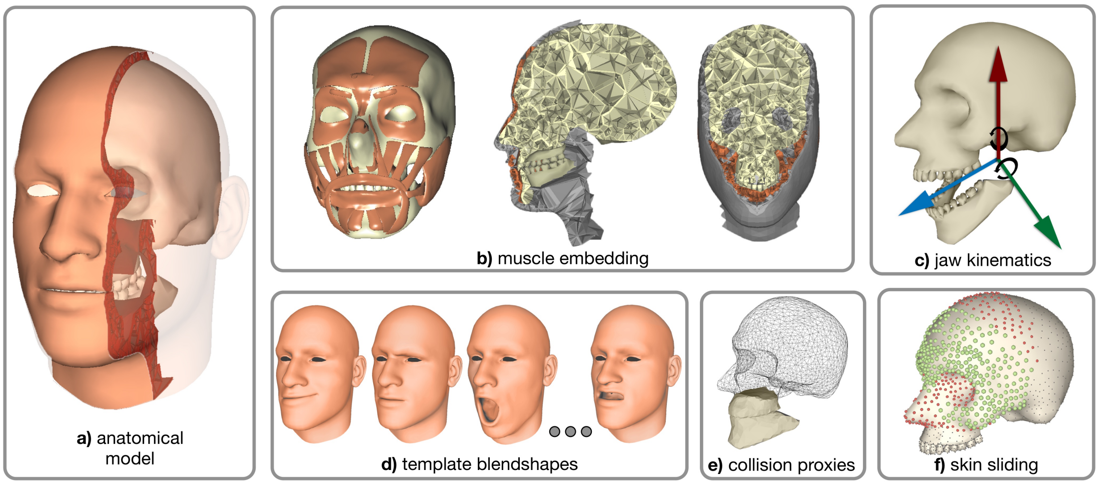


Figure 5.4: Our template model consists of a volumetric representation of the tissue and bones (a), and a surface blendshape basis to represent the expression space (d). Muscles are embedded into a non-conforming tetrahedral mesh discretization (b). We explicitly model jaw kinematics with a 5 DoF joint (c) and utilize low-resolution geometry proxies for faster collision detection for the teeth region (e). Dynamic skin sliding is supported by introducing both sliding (green) and fixed (red) constraints for bone-tissue connections (f).

Jaw kinematics. The relative motion of the jaw with respect to the skull contributes significantly to the final articulation of the face. The kinematics of the temporomandibular joint is non-trivial, consisting of both rotational and translational motion. In our model (see Figure 5.4-c), we define the major rotation axis (x-axis, corresponding to mouth opening) as the axis passing through the centers of the mandibular condyles. Halfway through the condyles, we define a perpendicular axis (y-axis) corresponding to vertical jaw rotation. The jaw does not normally rotate about the third orthogonal axis (z-axis), but it can translate (slightly) in all three directions. This amounts to 5 DoFs for the jaw motion, expressed with respect to the skull, which is treated as a free rigid body (our model does not include the craniocervical junction). We concatenate the kinematic parameters of the jaw bone into a vector $\vec{\mathbf{b}} \in \mathbb{R}^5$.

Template blendshapes. Given an anatomical model of the face, a natural control interface would be activation signals for all motor units. While biologically meaningful, such controls would not be user-friendly, because many motor units can affect a surface point in a complex, non-linear way. Instead, we augment our template model with a set of 48 blendshapes inspired by FACS Ekman and Friesen [1977] that have been sculpted by an artist on our generic face model. These blendshapes are only defined on the skin as a basis for parametrizing the space of facial expressions. They provide no information about the internal deformations, which are handled by physics-based simulation (Section 5.3.2 and Section 5.3.3). This combination of surface blendshape basis and volumetric simulation model allows us to retain compatibility with commonly used blendshape controls, while offering the benefits of advanced physics-based simulation effects.

5.3.2 Forward skinning model

The goal of the forward physics algorithm is to compute the deformed soft tissue and resulting skin surface given bone kinematics and muscle activation parameters. We model the latter with a vector $\vec{\mathbf{a}}$ (see "Active tissue" below) that represents the amount of activation (contraction) of all facial muscles. Even though in reality the jaw motion is controlled by muscle activations (in particular the masseter muscle) our model assumes the bones are directly controlled kinematically and the muscle activations are used only to create the facial expressions.

At the heart of our method is a physics-based model of soft tissue elasticity including muscle activation. We define this model using linear finite elements on our tet-mesh adapted for a given subject. Let $\vec{\mathbf{x}}$ denote a vector stacking all degrees of freedom of the soft tissue, i.e., the 3D coordinates of all nodes.

Passive tissue. For passive tissue we define deformation energy

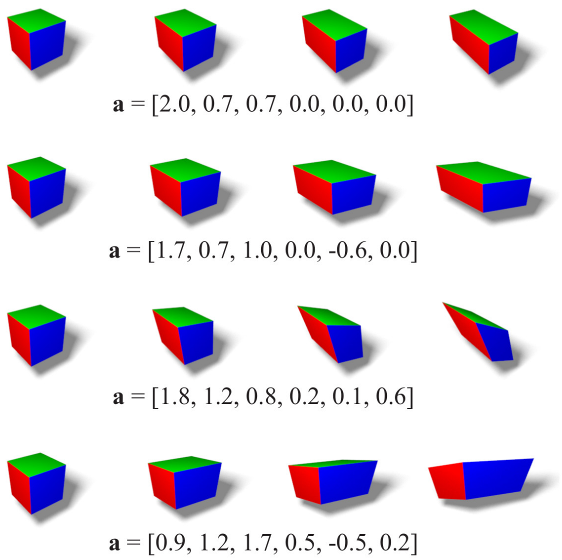
$$E_{\text{pass}}(\vec{\mathbf{x}}) = \sum_i \min_{\mathbf{R}_i \in SO(3)} W_i^{\text{pass}} \mu \|\mathbf{F}_i(\vec{\mathbf{x}}) - \mathbf{R}_i\|_F^2 + W_i^{\text{pass}} \lambda (\det(\mathbf{F}_i(\vec{\mathbf{x}})) - 1)^2, \tag{5.1}$$

where the index i goes over all tets and $W_i^{\text{pass}} \geq 0$ denotes the volume of the i-th tetrahedron that is occupied by passive tissue, pre-computed during template construction with Monte-Carlo sampling. The first term in Eq. 5.1 corresponds to the commonly used co-rotated elasticity (measure of deviation from rigid motion), while the second term models the resistance to changes of volume. $\mathbf{F}_i(\vec{\mathbf{x}})$ denotes the deformation gradient, and $\mathbf{R}_i$ is an auxiliary rotation matrix used in the co-rotated model Sifakis and Barbic [2012]. μ and λ are material parameters that we set by default to $\mu = 1$ and $\lambda = 3$. We can change these parameters to achieve specific effects as discussed in Section 5.3.6.

Active tissue. For tets corresponding to the active layer (muscles), we propose a novel activation model. Previous muscle models typically assume a given direction of muscle fibers along which the muscle contracts Teran et al. [2005a], Lee et al. [2009]. While this corresponds to the biological structure of muscles, the problem is that the exact muscle fiber directions are in general not known. Medical imaging techniques such as diffusion tensor imaging are prohibitively expensive and time consuming, and the signal quality is limited.

Previous work in graphics Saito et al. [2015] applied ad-hoc muscle fiber approximations which worked well for major skeletal muscles (such as the biceps), but are not sufficiently accurate for the delicate facial muscles. Along with the exact location of muscle insertion points, tuning of these parameters to obtain realistic facial expressions is possible, but tedious Sifakis et al. [2005]. To circumvent these issues, we propose a different muscle activation model that does not require explicit knowledge of fiber directions, but relies on the elementary bio-mechanical fact that muscles can generate only internal forces. In other words, an isolated muscle is not capable of translating or rotating by itself (even though the muscle can of course be translated or rotated due to contact with the surrounding tissues).

The property that the muscle cannot translate itself is already guaranteed by the translation invariance of the deformation gradient operator $\mathbf{F}_i(\vec{\mathbf{x}})$. Since a



a = [2.0, 0.7, 0.7, 0.0, 0.0, 0.0]

a = [1.7, 0.7, 1.0, 0.0, -0.6, 0.0]

a = [1.8, 1.2, 0.8, 0.2, 0.1, 0.6]

a = [0.9, 1.2, 1.7, 0.5, -0.5, 0.2]

Figure 5.5: Visualization of the capabilities of our 6-DoF activation model by squishing a cube, corresponding to a small sample of muscle tissue.

muscle tet should also not rotate itself, we require the activation to be a *symmetric* 3×3 matrix. Every symmetric matrix in $\mathbb{R}^{3\times3}$ has an eigendecomposition of the form $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathsf{T}$, where $\mathbf{Q} \in SO(3)$ and $\mathbf{\Lambda} \in \mathbb{R}^{3\times3}$ is diagonal. Therefore, the symmetric activation matrix corresponds to non-uniform scaling ($\mathbf{\Lambda}$) in an arbitrary orthonormal coordinate system ($\mathbf{Q}$). In other words, the symmetric matrix represents pure distortion without any change of orientation Shoemake and Duff [1992] (see Figure 5.5).

For each active tet, we define an activation vector $\vec{\mathbf{a}}_i \in R^6$ and use a linear operator $\mathcal{S} : \mathbb{R}^6 \to \mathbb{R}^{3\times3}$ to generate the corresponding symmetric matrix $\mathcal{S}(\vec{\mathbf{a}}_i) \in \mathbb{R}^{3\times3}$. Muscles, like most biological soft tissue, are approximately incompressible, which means that $\det(\mathcal{S}(\vec{\mathbf{a}}_i)) = \det(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\mathsf{T}) = \det(\mathbf{\Lambda})$ should be close to 1. However, to compensate for discretization errors, we do not enforce $\det(\mathcal{S}(\vec{\mathbf{a}}_i)) = 1$ strictly, but only as a soft constraint, as discussed in Section 5.3.3.

We use this activation model to define the deformation energy $E_{\text{act}}(\vec{\mathbf{x}}, \vec{\mathbf{a}})$ of active tissue, where $\vec{\mathbf{a}}$ is a vector stacking the 6-dimensional activation parameters for all active tets. Specifically, we define:

$$E_{\text{act}}(\vec{\mathbf{x}}, \vec{\mathbf{a}}) = \sum_i \min_{\mathbf{R}_i \in SO(3)} W_i^{\text{act}} \mu \|\mathbf{F}_i(\vec{\mathbf{x}}) - \mathbf{R}_i \mathcal{S}(\vec{\mathbf{a}}_i)\|_F^2 + W_i^{\text{act}} \lambda (\det(\mathbf{F}_i(\vec{\mathbf{x}})) - \det(\mathcal{S}(\vec{\mathbf{a}}_i)))^2, \tag{5.2}$$

where the index i goes over all tets and $W_i^{\text{act}} \geq 0$ represents the volume of the i-th tet that corresponds to active tissue. Here the co-rotated term aims to find the rotation $\mathbf{R}_i$ that best aligns the deformation gradient $\mathbf{F}_i(\vec{\mathbf{x}})$ with $\mathcal{S}(\vec{\mathbf{a}}_i)$. The second term encourages the volume ratio of the deformed tet (i.e., $\det(\mathbf{F}_i(\vec{\mathbf{x}}))$) to align with the volume ratio of the activation matrix $\det(\mathcal{S}(\vec{\mathbf{a}}_i))$, which should be close to 1, i.e., volume conserving.

Bone attachments. Muscles are connected to the bones using a complex network of connective tissue, whose exact function is a matter of active research

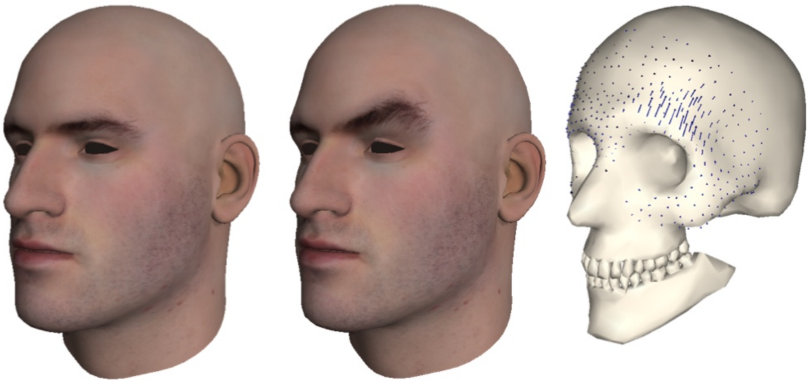


Figure 5.6: An eyebrow raise expression uses the skin sliding feature of our model. The blue arrows show the displacement of the contact vertices between the cranium and the flesh.

Schleip et al. [2013]. In animation, the visual importance of skin sliding is well recognized Li et al. [2013]. To distinguish areas where soft tissue is directly attached to the bones from areas where soft tissue slides over the bones, we create two types of constraints: 1) pin constraints and 2) sliding constraints. The pin constraints are straightforward to implement using Dirichlet boundary conditions. The sliding constraints are modeled as point-on-plane constraints on the tangent planes of the bone surfaces. We found this approximation to be sufficient even for curved regions, since the amount of sliding displacement is generally small.

Formally, we express both pin and sliding constraints using a function $\vec{\mathbf{c}}(\vec{\mathbf{x}}, \vec{\mathbf{b}})$ that depends also on the kinematic parameters $\vec{\mathbf{b}} \in \mathbb{R}^5$ of the jaw bone. All of the constraints are satisfied if and only if $\vec{\mathbf{c}}(\vec{\mathbf{x}}, \vec{\mathbf{b}}) = 0$. We have manually distributed the pin and sliding constraint as shown in Figure 5.4-f. The constraint types were chosen to achieve realistic deformations. For example, for an eyebrow raise expression, the skin slides over the skull as illustrated in Figure 5.6.

Quasi-static solution. In this section we discuss how to compute the quasi-static solution of the forward physics simulation, deferring the discussion of dynamics to Section 5.3.4. Quasi-statics means calculating a steady state where all dynamic motion has settled. The quasi-static regime is useful in generating static expressions and is particularly important when solving for muscle activations from observed shapes, as discussed in Section 5.3.3. Finding the steady state can be formulated as the following optimization problem:

$$\begin{aligned} &\underset{\vec{\mathbf{x}}}{\text{minimize}} \quad E_{\text{pass}}(\vec{\mathbf{x}}) + E_{\text{act}}(\vec{\mathbf{x}}, \vec{\mathbf{a}}) + E_{\text{grav}}(\vec{\mathbf{x}}) \\ &\text{subject to} \quad \vec{\mathbf{c}}(\vec{\mathbf{x}}, \vec{\mathbf{b}}) = 0, \vec{\mathbf{p}}(\vec{\mathbf{x}}) \geq 0, \end{aligned} \tag{5.3}$$

where $E_{\text{grav}}(\vec{\mathbf{x}})$ represents a linear gravitational potential (i.e., the familiar mgh). The inequality constraints $\vec{\mathbf{p}}(\vec{\mathbf{x}})$ are used to resolve penetrations (collision response) as follows. When collision detection finds a surface vertex penetrating the volumetric face model (see below for more details), an inequality constraint is appended to $\vec{\mathbf{p}}$. For each offending vertex we find its projection onto the surface and create a tangent plane at this point. The inequality constraint requires the vertex to be at the half-space opposite the volume.

We solve Eq. 5.3 by alternating between an interior point solver used to minimize Eq. 5.3 for fixed collision constraints $\vec{\mathbf{p}}$, and collision detection to update $\vec{\mathbf{p}}$. We have initially implemented a "homebrew" augmented Lagrangian solver, but ultimately decided to use the IPOPT package Wächter and Biegler [2006], which has proven to be more robust and usually needs fewer iterations to converge.

5.3.3 Inverse problem

The previous section explains how to compute face articulations for given bone positions and muscle activations. In this section we discuss the inverse problem. For a given target shape of the skin, we want to compute the corresponding bone parameters $\vec{\mathbf{b}}$ and muscle activations $\vec{\mathbf{a}}$, which, when used in the forward simulation (Eq. 5.3), will produce a skin surface close to the input shape.

Optimization formulation. Let $\vec{\mathbf{t}}$ denote the target vertex positions of the skin. The inverse modeling problem can be written as

$$\begin{aligned} \min_{\vec{\mathbf{x}},\vec{\mathbf{a}},\vec{\mathbf{b}}}. \quad & \|\mathbf{S}\vec{\mathbf{x}} - \vec{\mathbf{t}}^{\top}\|^2 \\ \text{subj. to} \quad & \vec{\mathbf{c}}(\vec{\mathbf{x}}, \vec{\mathbf{b}}) = 0, \vec{\mathbf{p}}(\vec{\mathbf{x}}) \geq 0 \\ & \nabla_{\vec{\mathbf{x}}} E_{\text{pass}}(\vec{\mathbf{x}}) + \nabla_{\vec{\mathbf{x}}} E_{\text{act}}(\vec{\mathbf{x}}, \vec{\mathbf{a}}) + \nabla_{\vec{\mathbf{x}}} E_{\text{grav}}(\vec{\mathbf{x}}) = 0 \end{aligned} \tag{5.4}$$

The objective term $\|\mathbf{S}\vec{\mathbf{x}} - \mathbf{T}\vec{\mathbf{t}}\|^2$ measures how close state $\vec{\mathbf{x}}$ is to the target $\vec{\mathbf{t}}$. The matrix $\mathbf{S}$ selects the simulation nodes corresponding to the skin surface. In addition $\mathbf{S}$ and $\mathbf{T}$ encode both position (point-to-point) and point-to-plane distance terms Rusinkiewicz and Levoy [2001]. The point-to-plane terms enable some amount of sliding (tangential motion) which is useful if we do not completely trust the correspondences represented by $\vec{\mathbf{t}}$. The last vector equality constraint describes the condition of quasi-static equilibrium, i.e., the sum of all forces (gradients with respect to $\vec{\mathbf{x}}$) is zero. Even though $\vec{\mathbf{x}}$ is also an optimization variable, the desired output are the optimal values of muscle activations $\vec{\mathbf{a}}$ and bone parameters $\vec{\mathbf{b}}$.

Numerical solution. As in Section 5.3.2, we use an interior-point method to solve the constrained optimization problem in Eq. 5.4. We alternate the interior point solverWächter and Biegler [2006] with collision detection that determines the non-penetration constraints $\vec{\mathbf{p}}$ as in Section 5.3.2.

Figure 5.7 shows an example of an inverse physics solve for two blendshapes of a user-specific blendshape model, visualizing separately the effect of the jaw motion and the effect of muscle activations.

5.3.4 Template-based Modeling and Animation

In this section we explain how we integrate the optimization algorithms presented above into a complete system for creating and animating subject-specific face simulation models.

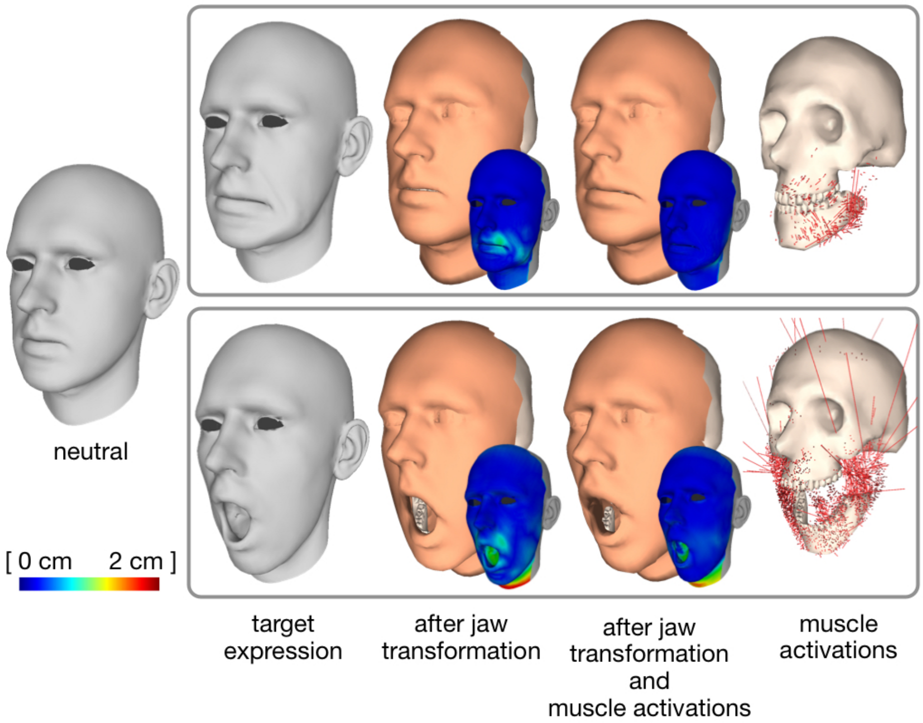


Figure 5.7: Inverse physics finds jaw transformation and muscle activations that accurately reproduce the target blendshapes.

Model Building. We start by 3D scanning the face of our subject in neutral expression and about 5-10 additional premeditated facial expressions using a multi-view stereo setup as described in Ichim et al. [2016]. Each of the scans is approximately aligned with the skin of our template model (Section 5.3.1) using rigid registration (plus uniform scale). Then we apply non-rigid ICP Rusinkiewicz and Levoy [2001] to find dense correspondences between the template skin and the target scan, guided with a few manually chosen markers as shown in the inset. We denote the registered skin surfaces as $\vec{\mathbf{s}}_{\text{neut}}$ for the neutral and $\vec{\mathbf{s}}_k$ for k-th expression.

Next, we deform our volumetric template model such that its boundary (skin) aligns with $\vec{\mathbf{s}}_{\text{neut}}$. This is accomplished with Anatomy Transfer Dicko et al. [2013], Ichim et al. [2016]. Note that during this process the generic face model can deform freely, i.e., the shape and/or volume of all cells can change, including the bones (in contrast to the deformation model considered in Section 5.3.2). We then use Example-Based Facial Rigging Li et al. [2010] to convert the registered expressions $\vec{\mathbf{s}}_k$ to subject-specific blendshapes $\vec{\mathbf{c}}_j$, $j = 1, \ldots, 48$.

The processing steps so far essentially rely on existing methods to align the volumetric template to the neutral expression and to create the subject-specific blendshape model. We refer to the above cited papers for implementation details on these algorithms. After this geometric preprocessing, we now solve for activations $\vec{\mathbf{a}}_j$ and jaw bone parameters $\vec{\mathbf{b}}_j$ that correspond to each of the blendshapes $\vec{\mathbf{c}}_j$ using the Inverse Physics optimization of Section 5.3.3.

Animation. To animate the created face model, we need to feed appropriate muscle activations and jaw bone parameters to the Forward Physics optimization of Section 5.3.2 for each animation frame. Given per-frame blendshape weights $\vec{\mathbf{w}} = \{w_1, \ldots, w_{48}\}$, we compute muscle activations as $\vec{\mathbf{a}} = \vec{\mathbf{a}}_{\text{neut}} + \sum_j w_j(\vec{\mathbf{a}}_j - \vec{\mathbf{a}}_{\text{neut}})$, where $\vec{\mathbf{a}}_{\text{neut}}$ corresponds to neutral activations, i.e., each activation $\mathcal{S}(\vec{\mathbf{a}}_{j,i}) = \mathbf{I} \in \mathbb{R}^{3\times3}$. Linear blending of the activation parameters is justified because there is no rotational component in symmetric matrices Shoemake and Duff [1992]. Similarly, we compute the blended jaw kinematics parameters $\vec{\mathbf{b}} = \sum_j w_j \vec{\mathbf{b}}_j$. While blending of rotation angles is in general not recommended, we found that for the limited range of rotations of the jaw this simple scheme does not produce any visible artifacts.

Dynamics. Adding inertia corresponds to a minor change of Eq. 5.3 as was discussed in Section 5.3.2. We use the popular backward Euler integration, which in its optimization form Liu et al. [2013] corresponds to augmenting the objective of Eq. 5.3 with the term: $\frac{1}{2}\|\vec{\mathbf{x}} - (\vec{\mathbf{x}}_n + h\vec{\mathbf{v}}_n)\|^2_{\mathbf{M}}$, where $\vec{\mathbf{x}}_n$ and $\vec{\mathbf{v}}_n$ are positions and velocities in the previous frame, $h > 0$ is the time step, and $\mathbf{M}$ is the mass matrix. We use a diagonal matrix $\mathbf{M}$ (mass lumping) with a soft tissue density of $1g/cm^3$. The minimizer $\vec{\mathbf{x}}$ of Eq. 5.3 then becomes the new state $\vec{\mathbf{x}}_{n+1}$ and the new velocity is $\vec{\mathbf{v}}_{n+1} = (\vec{\mathbf{x}}_{n+1} - \vec{\mathbf{x}}_n)/h$. The main difference from the quasi-static solution is that the dynamic solution depends on the previous state $(\vec{\mathbf{x}}_n, \vec{\mathbf{v}}_n)$, i.e., we need to execute the time steps in sequence. To add non-conservative external forces, such as wind, we proceed as in Projective Dynamics Bouaziz et al. [2014] and change the additional term to $\frac{1}{2}\|\vec{\mathbf{x}} - (\vec{\mathbf{x}}_n + h\vec{\mathbf{v}}_n + h^2\mathbf{M}^{-1}\vec{\mathbf{f}}_{\text{ext}})\|^2_{\mathbf{M}}$. Here $\vec{\mathbf{f}}_{\text{ext}} \in \mathbb{R}^3$ is the external force vector, e.g., a wind force is a function of triangle normal, area, and wind direction.

Plasticity. To support effects such as fattening or slimming, we use a standard model of plastic deformations. Specifically, each total deformation gradient $\mathbf{F}_{\text{total}}(\vec{\mathbf{x}})$ is assumed to be composed of an elastic deformation component and plastic deformation component, i.e., $\mathbf{F}_{\text{total}}(\vec{\mathbf{x}}) = \mathbf{F}_{\text{elast}}(\vec{\mathbf{x}})\mathbf{F}_{\text{plast}}$ or, equivalently, $\mathbf{F}_{\text{elast}}(\vec{\mathbf{x}}) = \mathbf{F}_{\text{total}}(\vec{\mathbf{x}})\mathbf{F}^{-1}_{\text{plast}}$. Note that $\mathbf{F}_{\text{plast}}$ does not depend on the current deformed state $\vec{\mathbf{x}}$. The deformation gradient $\mathbf{F}_i(\vec{\mathbf{x}})$ used in Eq. 5.1 and Eq. 5.2 corresponds to the elastic deformation component, because plasticity is a separate process, e.g., tissue growth, which is decoupled from elastic deformations. Therefore, the only modification we need to make to account for plasticity is to replace the $\mathbf{F}_i(\vec{\mathbf{x}})$ in Eq. 5.1 and Eq. 5.2 by $\mathbf{F}_i(\vec{\mathbf{x}})\mathbf{F}^{-1}_{\text{plast},i}$, where $\mathbf{F}_{\text{plast},i}$ describes the plastic deformation of the i-th tet. In our system, we use only uniform scaling, i.e., $\mathbf{F}_{\text{plast},i} = s_i\mathbf{I}$, where $s_i > 0$ is a scaling coefficient (corresponding to growth for $s_i > 1$ and shrinking for $0 < s_i < 1$). The settings of the s_i parameters for each tet depend on the effect we wish to achieve as discussed in Section 5.3.6. Plasticity, as well as inertia and external forces are applied in forward physics only.

5.3.5 Evaluation

Before showing application results of our method in Section 5.3.6, we evaluate the behavior of our optimization algorithms and provide comparisons to previous work.

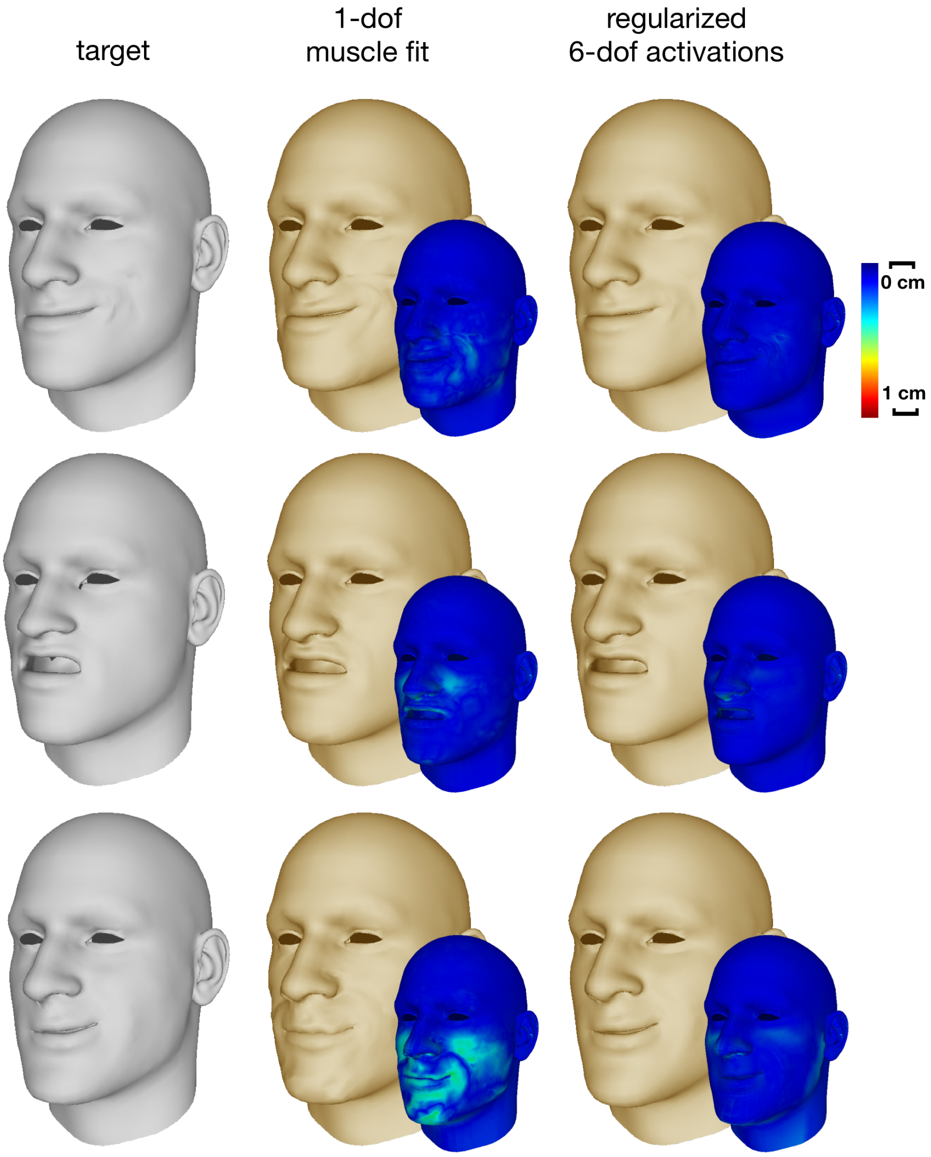


Figure 5.8: Our 6-DoF muscle activation model (rigt) leads to more accurate reconstruction of the target expression (left) than previous 1-DoF fiber-aligned activations models (middle).

Muscle activation model. As mentioned in Section 5.3.2, previous methods constrain the deformation along muscle fibre directions Teran et al. [2005a], Sifakis et al. [2005], Lee et al. [2009], Saito et al. [2015]. In our experiments we found that muscle fiber directions can be unreliable and lack the flexibility to accurately reproduce all facial expressions. This insight triggered the design of our more general activation model. In Figure 5.8 we compare the results of inverse physics with our method and the previous fiber-restricted model, where fiber directions are computed from our geometric muscle models using the method of Choi et al. [Choi and Blemker, 2013]. The active tetrahedra of the 1-DOF muscle model act based on the following constraint energy:

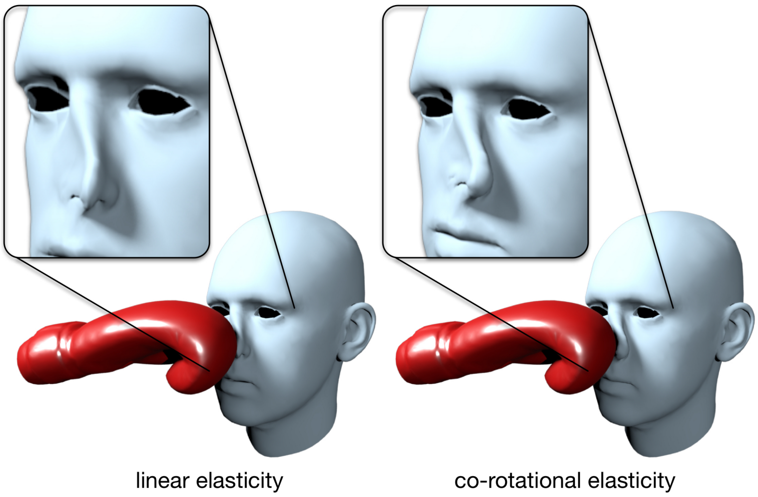


Figure 5.9: A boxing punch to the nose results in artifacts with an elastic model lacking rotation invariance as in Ichim et al. [Ichim et al., 2016] (left). More realistic deformations are obtained with our rotation-invariant model (right).

$$E_{act\_1DOF}(\vec{\mathbf{x}}, \vec{\mathbf{a}}) = W_i^{\mathrm{act}} \mu \|\mathbf{F}_i(\vec{\mathbf{x}}) - \mathbf{R}_i \mathbf{Q}^T \mathcal{S}_{1DOF}(\vec{\mathbf{a}}_i)\mathbf{Q}\|_F^2 + \\ W_i^{\mathrm{act}} \lambda (\det(\mathbf{F}_i(\vec{\mathbf{x}})) - \det(\mathcal{S}(\vec{\mathbf{a}}_i)))^2,$$

where $\mathbf{Q}$ encodes the muscle fiber orientations and $\mathcal{S}_{1DOF}(\vec{\mathbf{a}}_i) = diag(\vec{\mathbf{a}}_i, 1, 1)$. As Figure 5.8 illustrates, muscle activations constrained to the fiber directions fail to closely match the desired target shape, while our activation model leads to a much more accurate reconstruction of the target expression.

Comparison to volumetric blendshapes. Defining a deformation model that is invariant under rigid motions is essential for correct tissue behavior. The volumetric blendshape approach of Ichim et al. [Ichim et al., 2016] lacks rotation invariance, which can lead to artifacts, e.g., when large rotation of the soft tissues are induced by external forces, such as the boxing punch shown in Figure 5.9. We propose rotation-invariant models for both passive and active soft tissue, leading to more realistic results. While we distinguish between passive and active tissue, previous work Ichim et al. [2016] assumes that all soft tissue can activate. In addition, our approach includes a kinematic model for the jaw, whereas Ichim et al. [Ichim et al., 2016] only approximated the jaw by using a more stiff (but not exactly rigid) material. Finally, our method also allows for skin sliding, facilitating more realistic flesh deformations especially in areas such as the forehead (Figure 5.6).

Model adaptations. Our approach supports animating a character after significant modifications of the neutral pose (e.g. slimming/fattening, bone modifications, see Section 5.3.6) using the same muscle activation patterns. One might argue that the same effects could be obtained by using deformation transfer Sumner and Popović [2004] on traditional linear animation models. For example, similar

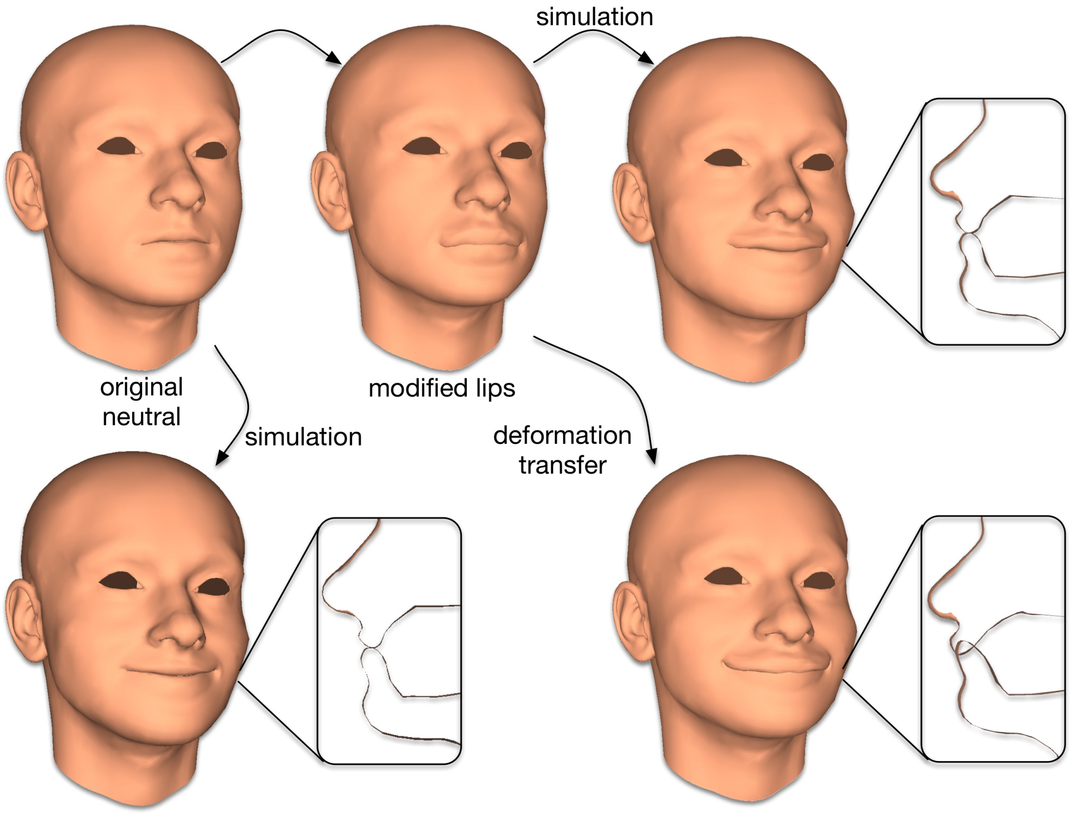


Figure 5.10: Model adaptations such as increased lip volume are handled accurately in our approach, while deformation transfer Sumner and Popović [2004] leads to self-intersections.

modifications as the ones we propose could be applied on the surface mesh of the neutral blendshape. Deformation transfer on all expression blendshapes will then yield new face rig that incorporates the desired changes. However, this approach has the significant drawback that the new blendshapes are not necessarily consistent with the same blendshape weights, e.g., self-intersections easily occur as shown in Figure 5.10.

In addition, direct transfer of modifications to the neutral pose cannot account for the complex force interactions in the elastic tissue. For example, when increasing the volume of the lips, the expression dynamics will change as a consequence of the changed stress distribution. Our indirect approach, that solves for the facial pose given muscle activations, can accommodate such scenarios and leads to more natural expressions.

Statistics. The interior point solver of the forward physics optimization requires on average 8 iterations per frame to converge. This takes approx. 22 seconds including the collision detection update on a consumer laptop with a 3.1 GHz Intel Core i7 processor and 16GB of main memory. The inverse problem needs approx. 15 iterations to compute the jaw transformation and muscle activations, averaging at about 3 minutes per target shape. The volumetric face template model of the passive flesh and active muscles used for the results presented in this project has 8098 vertices and 35626 tetrahedra. The active muscle layer covers

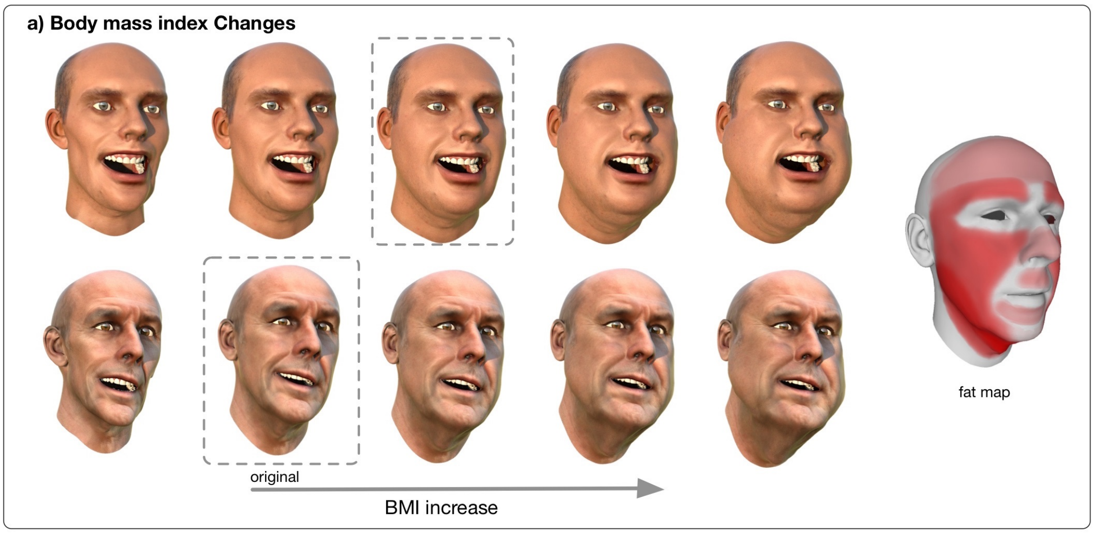


Figure 5.11: Application Demos I: a) Body mass index changes and their impact on expressions. The original avatar is highlighted with dashed lines. More intense red in the fat map means more volume change of the corresponding face region.

approx. 27% of the entire flesh. The surface mesh model of the entire skin has 6393 vertices and 12644 faces.

5.3.6 Application Demos

We present a series of application demos to highlight the versatility of our approach. A key benefit of our physics-based simulation is that we can modify the static and dynamic parameters of the model to achieve a number of advanced animation effects that would be difficult to obtain with purely generative geometric methods. Please also refer to the accompanying video A.2 to better appreciate the dynamics of the animations.

All animation examples were driven by a temporal sequence of blendshape weights obtained from the performance capture system of Weise et al. [2011]. The tracking software also provides a rigid body transformation $\mathbf{T} \in SE(3)$ corresponding to the global rotation and translation of the head, as well as pitch and yaw for each of the eyeballs, which are parented to the head transformation $\mathbf{T}$.

Body mass index changes. Figure 5.11-a illustrates how an animated avatar can be modified to slim or fatten the person's face by adapting the plasticity scale for the soft tissue tets. As this adaptation alters the face geometry, simply re-animating the blendshape model would lead to unnatural expressions and visual artifacts caused by self-intersections. Our simulation approach avoids self-collisions and balances the stress distribution in the facial tissue while preserving the actuation forces, which leads to more plausible expressions and natural dynamics.

To create the scaling parameters $s_i > 0$, we start from a surface "fat map" painted by the user that specifies which areas of the face are more prone to fat accumulation. The values of the fat map are propagated into the volumetric

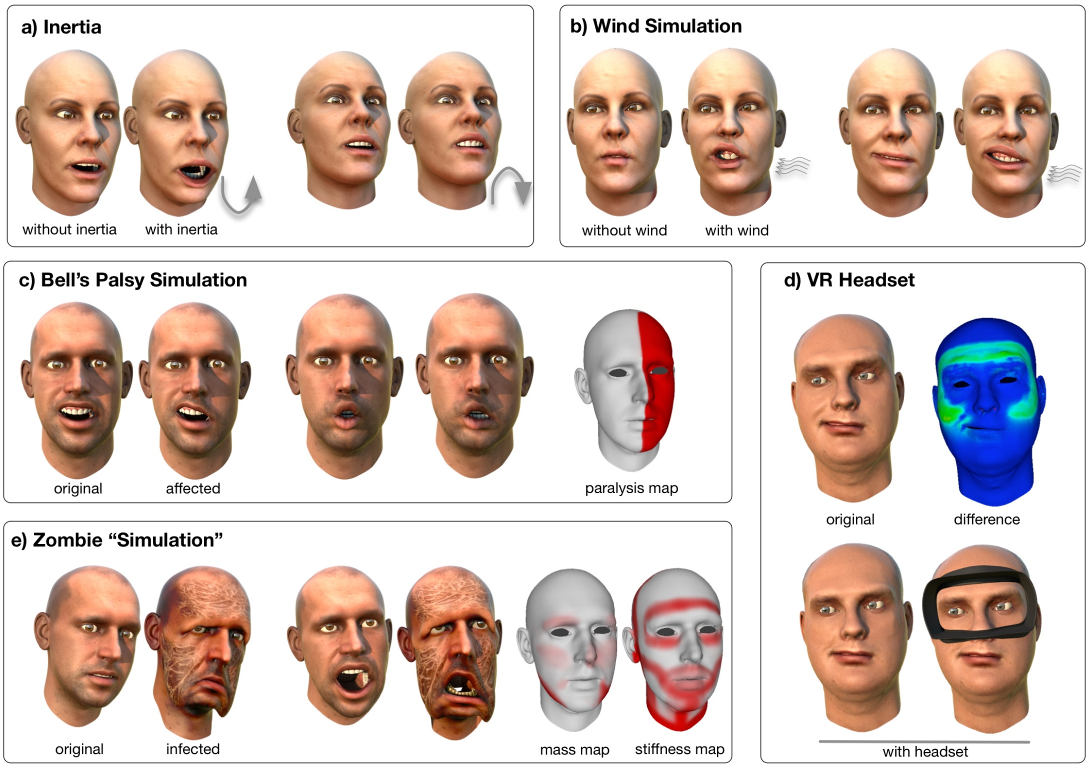


Figure 5.12: a) Simulating inertia under sudden motion changes (e.g., jumping). b) Dynamic deformations in a wind force field. c) Simulating Bell's Palsy affecting half of the face of an actor. d) VR headset obstructing the full motion of expressions on the face. e) Artistic editing to create a zombie character by adapting the mass and stiffness distribution as indicated in the color-coded maps.

tet-mesh by a diffusion process, similar to standard polygon-mesh diffusion flow (Botsch et al. [2010], Chapter 4.2), but using the volumetric Laplacian instead of the surface Laplace-Beltrami. We apply forward Euler integration with time step and number of steps adjusted by the user in an interactive graphical tool to achieve the desired volumetric propagation effect. We used the same fat map for both characters in Figure 5.11-a, uniformly scaled to achieve slimming or fattening. To account for the increased fat content in the soft tissue, we lower the stiffness μ to $0.8, 0.5, 0.3$ for the three levels of fattening shown in Figure 5.11-a. For slimming, we keep the default stiffness $\mu = 1$.

Inertia. Figure 5.12-a shows how our method incorporates inertial deformations in the dynamic simulation. Such secondary motion becomes particularly important in animations with strong accelerations, such as jumping, head shaking, or boxing.

Interaction with external forces and objects. Figure 5.12-b shows how an animation can be augmented with complex external force interactions produced by a dynamic wind field. Figure 5.12-d illustrates how a speech animation is affected when the subject is wearing a VR headset. Our contact resolution method adapts the face deformations to account for the collisions with the headset, creating non-linear bulging and wrinkling effects due to volume preservation of the facial tissue.

Simulation of muscle paralysis. In Figure 5.12-c, we show how muscle activations can be modified to simulate Bell's palsy syndrome, where the affected person is unable to activate certain facial muscles. In this example, we marked the active muscles of the left half of the face to behave like passive tissue, which simulates the effect of partial facial paralysis.

Extreme face modifications. To push the limits of facial modifications, we created a virtual zombie character in Figure 5.12-e. We designed two texture maps to modulate the mass and stiffness (see Figure 5.12-e) and extrapolated their values into the volume using our diffusion tool. The idea was to increase the mass of the cheeks to create a flesh sagging effect, while increasing stiffness around the lips and the eyes to avoid excessive pulling of the flesh. The final μ values vary between $0.7 - 5.7$ and the density varies between $1 - 3g/cm^3$, achieving artistic "undead" effects.

5.3.7 Limitations and Future Work

In our approach we rely solely on a generic volumetric template and a set of surface scans of the modeled person to derive the interior facial structure. This inherently limits the accuracy of our approach in terms of the true facial dynamics of the scanned actor. Getting access to the internal structure through volumetric scanning devices would allow building more faithful simulation models, but incurs a high acquisition cost. We follow this path in the Section 5.4.

Another approach for future work is to build a statistical model of the bone and tissue structures from a sufficiently large set of volumetric scans, similar to the morphable face models that have been successfully applied for the skin surface Blanz and Vetter [1999].

Detailed physical simulation is computationally involved and our method is currently not suitable for realtime animation. While computational efficiency was not the main focus of our work, we believe that significant speedups can be achieved, in particular by more explicitly exploiting spatial and temporal coherence. In the context of realtime animation, our approach could potentially be used to automatically create corrective shapes for a given blendshape basis in an offline process. How to select an optimal set of such correctives based on a given simulation is an interesting avenue for future research.

Our tet-mesh discretization is currently too coarse to correctly model small-scale effects such as skin wrinkles. However, increasing the resolution to the appropriate scale would lead to prohibitive computation times. Therefore, in future work, we want to explore ways to combine our simulation model with procedural or data-driven methods for wrinkle generation to further increase the visual realism of the animations.

Other avenues for future work include modeling and simulating hair, adding person-specific teeth models and a simulation of the tongue.

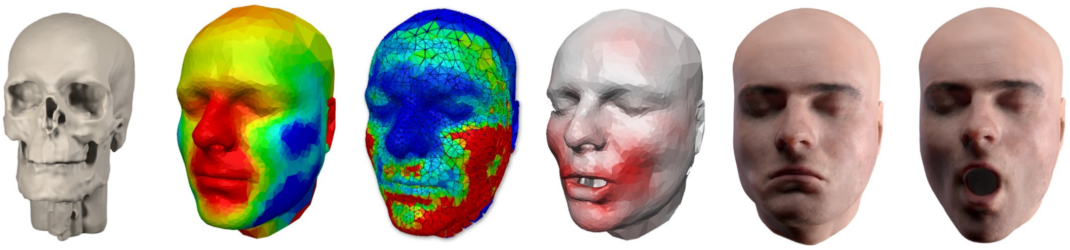


Figure 5.13: We build a static anatomical face model from the MRI and use 3D surface scans as training data to learn mechanical parameters that explain deformations of the real face using physics-based simulation.

5.4 Learning mechanical properties and muscle control

One of the difficulties in simulating a realistic facial animation is modeling of underlying anatomy. The recent previous work Cong et al. [2015], Ichim et al. [2016], Lan et al. [2017] and the work presented in the Section 5.3 adapted a template model by deforming template rigid bone structure to fit a personalized neutral surface mesh Dicko et al. [2013]. This step, however, possibly introduces a large error in the anatomy which might be a crucial roadblock in the following steps of fitting delicate facial expressions. Large variability in muscle shapes and locations are found in cadaveric studies [Pessa et al., 1998], many people might even be lacking some muscles or they might be present only on left/right side, e.g. Risorius muscle or Nasalis muscle [Waller et al., 2008]. The whole internal anatomical structure including connective tissues (i.e. fascia, ligaments, tendons) differ largely within subjects. All those factors influence the resulting shape of the face.

In this section, we propose a new approach to creating realistic physics-based face models, aiming to bridge the gap between graphics and biomechanics. First, we designed a simple synthetic experiment shown in Figure 5.14 to determine whether we can learn complex material properties of a cylinder affected by known external forces and observations of its surface deformations. The results showed (Figure 5.14 c) that even a single observation can be used for a non-trivial material parameter fitting. This sparked the idea of using various observations of facial soft tissue to discover its complex structure. We start by building a static face model in a neutral expression from an MRI scan. Note that MRI is not essential for our method and any template model adapted Dicko et al. [2013] to the neutral scan can be used with some loss of accuracy depending on the distance of the adapted template model from the subject-specific skull geometry as shown in Section 5.4.5. We generate our segmentations and tet-meshes using existing tools Fedorov et al. [2012] and claim no contributions in this part. Our main contribution consists of automated learning of mechanical properties of our subject's face from data (3D scans of facial deformations).

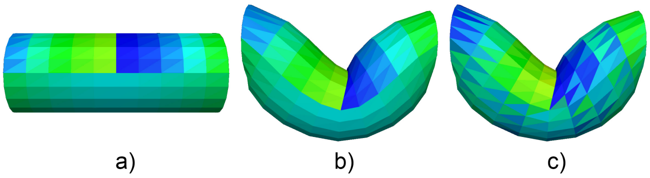


Figure 5.14: Synthetic experiment for learning material parameters. a) Tetmesh in a rest pose with color coded varying stiffness not affected by gravity. b) The mesh in quasi-static equilibrium fixed on sides with applied gravity force c) Our prediction of stiffness parameters given the rest pose and surface of the tetmesh in b)

An interesting fact attributable to the high elasticity of the face is that changing the direction of the gravity leads to surprisingly large deformations Ozsoy et al. [2015]. By varying the direction of the gravity (by posing the subject's head in various orientations) and by observing various facial expressions created by voluntary muscle activations, we can deduce the composition and mechanical properties of facial soft tissues. To be able to explain our captured facial deformations, we found that in addition to realistic muscle activation models, it is important to use heterogeneous elastic properties including prestrain. Prestrain refers to a biomechanical model of rest-pose stresses of biological tissues Gee et al. [2010], analogous to the tension in the strings of a guitar. Adding prestrain introduces anisotropic trainable parameters which improves the explanatory power of our model while being well motivated biologically. Our method does not require any manual segmentation of individual muscles, nor any prior knowledge of muscle fiber directions of a template model.

5.4.1 Static Anatomical Model

Building a geometric model of facial anatomical structures is non-trivial even if we consider only a static neutral facial expression and do not require segmentation of indvidual facial muscles. To avoid the laborious anatomical modeling process, some recent work starts from high-quality commercial anatomical models, such as Zygote Body Ichim et al. [2016, 2017]. A drawback is Zygote's licensing policy, which restricts re-distribution of any derived 3D models (re-distribution of images and videos is not restricted). In this project, we choose to build a brand new anatomical model of the face directly from MRI scans.

MRI data processing

We started by capturing four MRI scans of a neutral expression of one subject derived from two sequences (MP2RAGE INV1/INV2/UNI and T2, see Figure 5.15) and processed them using 3D Slicer Fedorov et al. [2012]. Specifically, we applied Bias Field Correction to eliminate intensity variations and then denoised the data using Gradient Anisotropic Diffusion filters. We proceeded with segmentation of soft and hard tissues using thresholding and region growing. Note that segmentation of the skull and the mandible from MRI is a challenging task because bone tissue does not produce much signal and it is therefore easy to confuse with air, which can be problematic in areas such as the sinuses.

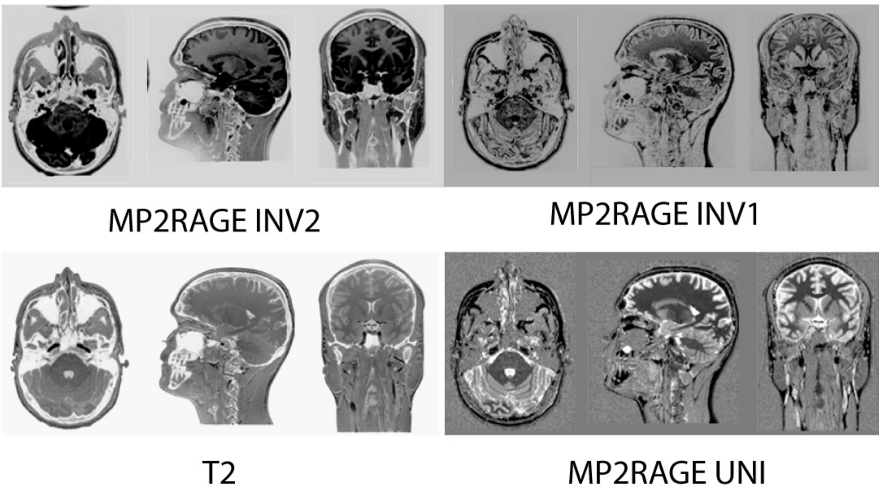


Figure 5.15: Examples of slices from our MRI sequences used to build our model.

To address these challenges we tried modern MRI sequences better suited for solids, specifically, ultrashort echo time (UTE) Robson and Bydder [2006]. The signal strength on bone tissue improved, but the images were very blurry. Therefore, we instead applied a "pseudo-CT" approach Torrado-Carvajal et al. [2016] which is a data-driven technique to convert MR images into CT-like images (we note that segmenting bones from CT scans is easy and highly accurate, however, we avoided direct CT scanning because it would expose the subject to radiation). Our resulting segmentations of the bones and skin can be seen in Figure 5.16. When finished with 3D image processing, we used the marching cubes algorithm Lorensen and Cline [1987] to create triangle meshes corresponding to the outer surface of the head (the skin) and the outer surfaces of the bones (the skull and the mandible). The triangle meshes were re-meshed using Meshmixer Schmidt and Singh [2010] to improve mesh quality and used as constraints in TetGen Si [2015] which produced our final tet-meshes, see Figure 5.17.

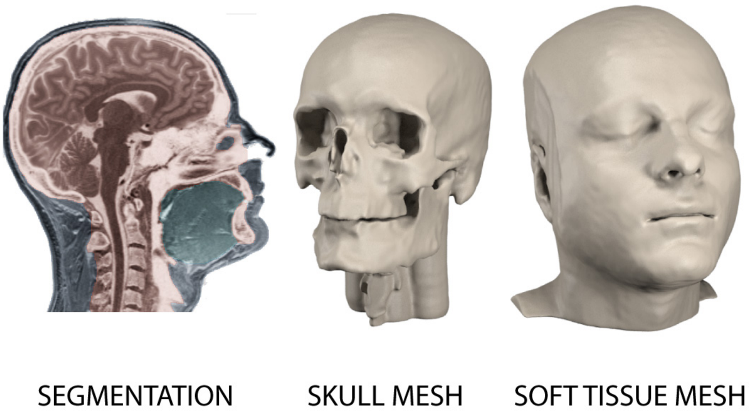


Figure 5.16: Segmentation of soft tissues and the bones from our MRI scans.

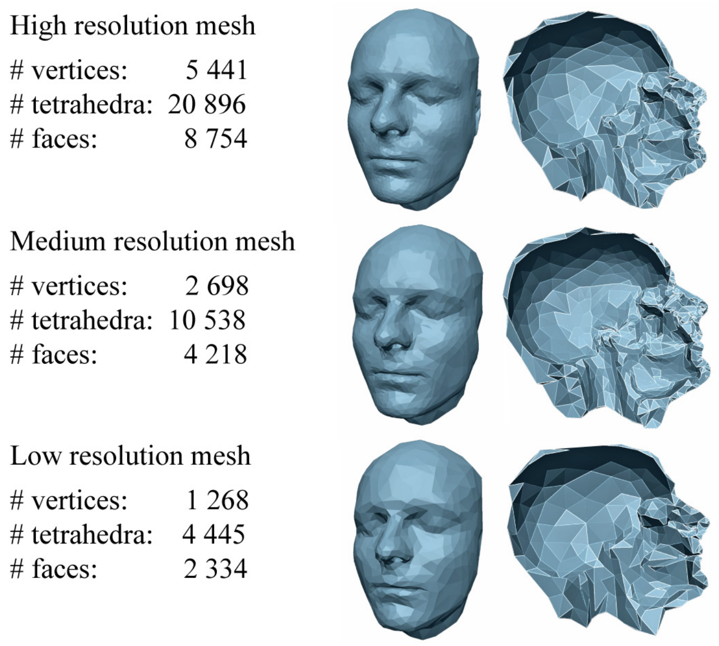


Figure 5.17: Our tet-meshes with varying resolution.

Surface 3D scans

Even though modern MRI scanners are very powerful and provide good resolution volumetric scans of the human body without any radiation, they have limitations. In addition to the high cost (around $500/hour), long scanning times and limited availability of MRI machines, an important limitation is that the subject must remain motionless inside the MRI scanner for several minutes (depending on the sequence). This means that MRI scanning of most facial expressions is practically impossible because muscle fatigue would prevent the subject from staying still for minutes. Important advances in Real-time MRI have been made in recent years Zhang et al. [2014], however, the methods are generally limited to either 1) very low-resolution volumetric imaging or 2) high-resolution single-slice images; neither is adequate for our purposes.

Instead, we captured geometry of deformed facial shapes using a structured light scanner (Artec Spider), producing detailed 3D scans of the skin. Specifically, in our deformed facial shapes we vary gravity directions (by changing the subject's head orientation) and facial expressions (by asking the subject to smile, frown, etc., see Figure 5.18). The face is quite supple and varying gravity directions results in surprisingly large skin displacements, see Figure 5.23.

Composition. A necessary pre-requisite for subsequent automatic parameter fitting (Section 5.4.3) is registration: finding correspondences between the captured 3D scans. To do this, we first register the tet-mesh we built from the MRI scan with a 3D scan of a neutral expression in the supine pose, i.e., similar setup as in the MRI scanner where the subject is reclining on a motorized patient table. The two shapes (MRI and 3D scan) are close, but not identical due to geometric distortions of MR images. Geometric distortion is a well-studied problem Baldwin et al. [2007] and its corrections are standard, however, despite all efforts some geometric distortion still remains. We deform our tet-mesh to match the 3D scan using non-rigid Iterative Closest Points (ICP), enabling small deformations of the

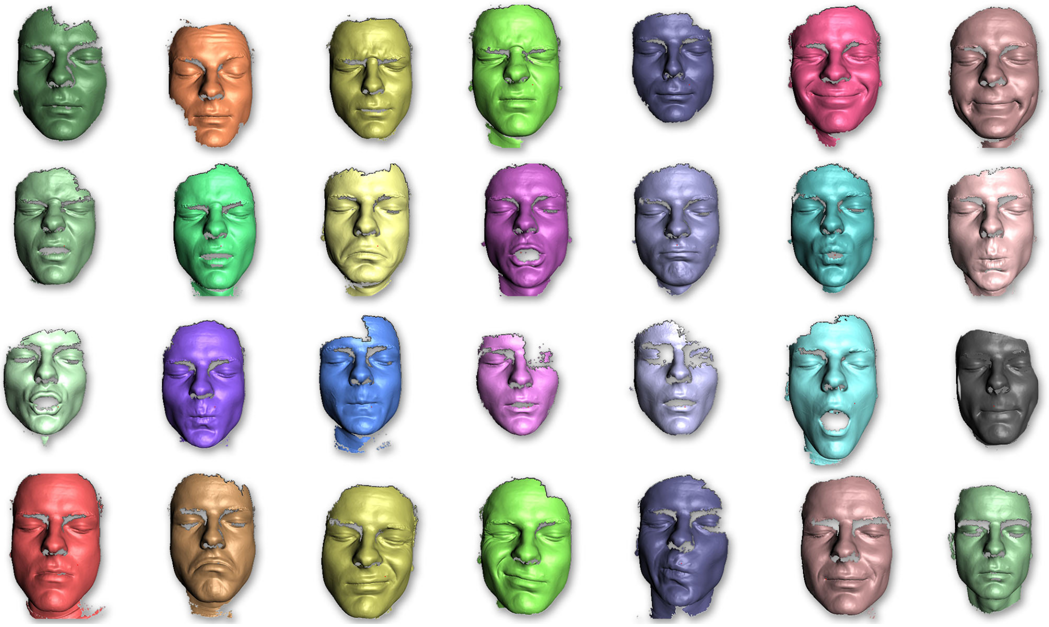


Figure 5.18: Example of facial expressions scanned using a portable structured light scanner.

entire mesh including the bones to compensate for the geometric distortions of the MRI. When finished, the surface of the tet-mesh is closely aligned with the 3D scan, allowing us to transfer the albedo map from the 3D scan to the surface of our tet-mesh. The albedo map with painted markers (see Figure 5.19) helps us to find correspondences among our set of 3D scans of deformed facial shapes. Specifically, we apply another non-rigid ICP process with normalized cross-correlation of image patches as a data term Beeler et al. [2011] and volumetric deformation of the tet-mesh as a regularization term Ichim et al. [2016], this time assuming the skull and the jaw are rigid, because the shapes of the bones must be the same in all physiologically deformed facial shapes. Finding the rigid transformation of the skull in each of the shapes gives us a “rigid pre-stabilization”, i.e., an estimate of compensation for global head motion Beeler and Bradley [2014]. This is only an initial guess (hence the term “pre-stabilization”) which will be subsequently refined when solving our inverse modeling problem (Section 5.4.3). The output of this preparatory phase is a tet-mesh discretizing facial soft tissues which is accurately registered with our 3D scans of facial deformations. These data will be used to train our mechanical model of the face.

The main technical challenge we address is an inverse problem where we solve for mechanical parameters of a model that will match acquired 3D scans of real facial shapes under various external loads and muscle activations. Solving it leads to a large optimization problem which can not be solved separately per expression.

5.4.2 Mechanical model

We start by discussing the key mechanical modeling concepts used in our method. Building upon the finite element method Sifakis and Barbic [2012], we are optimizing for a rest shape $\vec{\mathbf{x}}_{\text{rest}}$ in a static equilibrium subject to no external forces (including gravity). The rest shape can, however, be subject to internal forces rep-

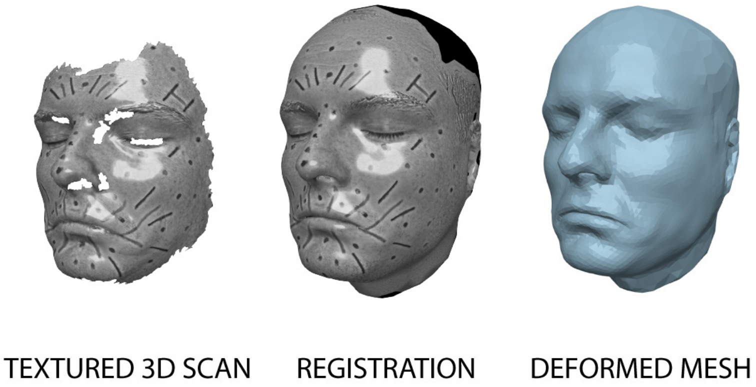


Figure 5.19: An example of registration of textured 3D scans.

resenting inherent residual biological strain which in addition to varying material parameters μ, λ affect the behavior of soft tissue elastic deformation.

Modeling passive tissue. When excised from the body, biological soft tissues retract, i.e., release stored elastic potential energy. This mechanical effect can be modeled with prestrain and it plays an important role in the body, e.g., in stabilizing joints Ellis et al. [2006]. We use linear finite elements and model prestrain by introducing a material space-dependent function $\mathbf{G} \in \mathbb{R}^{3\times 3}$ which modulates the deformation gradient. The total internal elastic potential energy of a prestrained deformable solid can be approximated as:

$$E_{\text{elast}}(\vec{\mathbf{x}}) = \sum_{i=1}^{m} v_i \Psi(\mathbf{F}_i(\vec{\mathbf{x}}, \vec{\mathbf{x}}_{\text{rest}})\mathbf{G}_i)$$

where $v_i > 0$ is rest-pose volume of i-th tetrahedron, Ψ is a hyperelastic energy density function, $\mathbf{F}_i \in R^{3\times 3}$ is deformation gradient (function of the current deformed state $\vec{\mathbf{x}} \in \mathbb{R}^{3n}$ and the rest pose $\vec{\mathbf{x}}_{\text{rest}} \in \mathbb{R}^{3n}$) and $\mathbf{G}_i \in R^{3\times 3}$ is the prestrain for i-th tetrahedron (m is the number of tets and n the number of vertices of our tet-mesh). With $\mathbf{G}_i = \mathbf{I}$, E_{elast} becomes equivalent to a standard FEM setup Sifakis and Barbic [2012]. With general $\mathbf{G}_i$, there can be non-zero potential energy stored in the configuration $\vec{\mathbf{x}}_{\text{rest}}$, i.e., $E_{\text{elast}}(\vec{\mathbf{x}}_{\text{rest}}) > 0$, making the naming convention less intuitive. Observing the object configuration $\vec{\mathbf{x}}_{\text{rest}}$ without prestrain or measuring the prestrain directly is difficult. One would have to make infinitely small dissections corresponding to each material point of our model Maas et al. [2016]. An observable rest pose $\vec{\mathbf{x}}_{\text{rest}O}$ affected by prestrain and gravity is a solution to $\nabla E_{\text{elast}}(\vec{\mathbf{x}}_{\text{rest}O}) + \nabla E_{\text{grav}}(\vec{\mathbf{x}}_{\text{rest}O}) = 0$, subject to boundary conditions (e.g., fixed skull vertices).

Modeling muscles. Facial expressions are created by contractions (activations) of facial muscles. Muscle tissues have complicated anatomy, but the basic building block are myocytes – long tubular cells which are able to contract and thus generate tension.

Unfortunately, capturing the fiber direction of thin facial muscles is a difficult task even with diffusion tensor imaging Levin et al. [2011]. The 3D surface of facial expressions are very sensitive to the directions of contractile muscle forces, which has been identified to be a major obstacle in realistic facial modeling Cong et al. [2016].

Recent work addressed this challenge with artist-directed muscle activations Cong et al. [2016] or by flexible muscle activation models as in Section 5.3. Specifically, the activation model in Section 5.3 assumes that in each facial expression, each muscle tissue element (tetrahedron) can deform itself arbitrarily in six dimensions (corresponding to a 3×3 symmetric matrix). Even though this model correctly disregards translational and rotational motions, it is still overly permissive, because it assumes that muscle activations can change arbitrarily among different facial expressions, ignoring the existence of muscle fibers and their contractile action. Muscle activations should not be controlled by six degrees of freedom, but only one which corresponds to neural control signal.

To address these issues, we propose the following muscle-activation function:

$$\mathcal{A}(\alpha_i, \mathbf{D}_i) = (1 - \alpha_i)\mathbf{I} + \alpha_i \mathbf{D}_i \tag{5.5}$$

where $\alpha_i \in [0, 1]$ is time-dependent activation of i-th element (tet) and $\mathbf{D}_i \in \mathbb{R}^{3\times3}$ is time-*independent* maximal deformation due to muscle contraction (and $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity). An important difference from the muscle model used in Section 5.3 is the fact that *only* α_i are allowed to vary between facial expressions; $\mathbf{D}_i$ is a subject-specific variable representing anatomy and thus is the same for all facial expressions.

This leads to a computationally more complicated inverse problem, because all facial expressions (smile, frown, ...) are now coupled in one large optimization problem, whereas in Ichim et al. [2017] the problem was separable per expression.

To build as accurate of a model as possible, we need to clearly differentiate between deformation due to internal passive, active and external forces. Compared to the model described in Section 5.3, we achieve this by solving simultaneously for both passive and active deformation properties. This is important even for neutral expressions. Even though a subject can be relaxed, some muscles are unconsciously activated (e.g. pushing lower teeth against upper teeth). A large difference in neutral expression can be observed in persons affected by Bell's palsy.

Elastic deformation. Our internal elastic potential uses corotated linear elasticity Sifakis and Barbic [2012] augmented with prestrain and a muscle activation model:

$$\begin{aligned} E_{\text{elast}}(\vec{\mathbf{x}}, \mathbf{A}, \mathcal{H}) = \sum_{i=1}^{m} \quad & v_i \mu_i \|\mathbf{P}_i(\vec{\mathbf{x}}, \vec{\mathbf{x}}_{\text{rest}}, \mathbf{G}_i) - \mathcal{A}(\alpha_i, \mathbf{D}_i)\|_F^2 \\ & + v_i \frac{\lambda_i}{2} tr^2(\mathbf{P}_i(\vec{\mathbf{x}}, \vec{\mathbf{x}}_{\text{rest}}, \mathbf{G}_i) - I) \end{aligned} \tag{5.6}$$

The vertex positions $\vec{\mathbf{x}} \in \mathbb{R}^{3n}$ and muscle activations $\alpha \in \mathbb{R}^m$ are time-varying parameters (where n is the number of vertices, m the number of tets in our mesh and $\mathcal{A} \in \mathbb{R}^{3x3}$ is a symmetric matrix). The notation $\mathcal{H} = (\vec{\mathbf{x}}_{\text{rest}}, \mathbf{E}, \mathbf{G}, \mathbf{D})$ collects all time-invariant, subject-specific parameters: the rest pose $\vec{\mathbf{x}}_{\text{rest}}$, Young's modulus $\mathbf{E}$, prestrain $\mathbf{G}$ and maximal muscle deformations $\mathbf{D}$. While constant in time, these parameters vary spatially, as indicated by the tetrahedron index i.

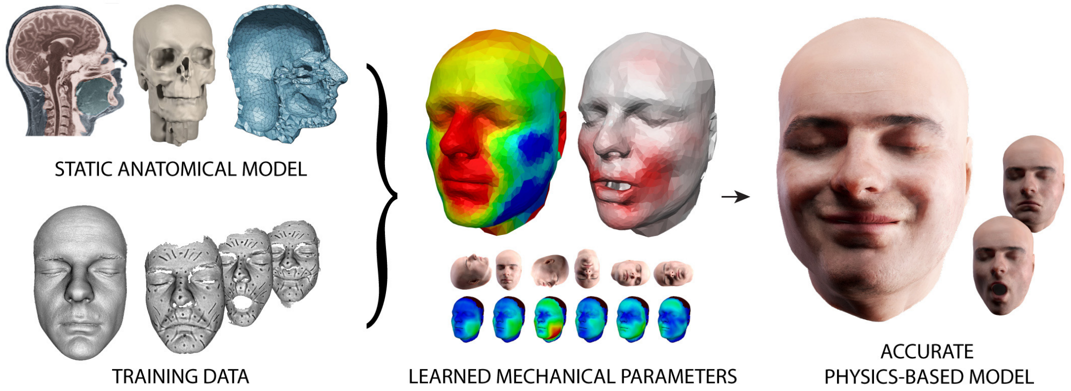


Figure 5.20: Given a static face model, we use 3D surface scans as training data in our inverse modeling problem. The result is a physics-based mechanical model with material parameters that explain deformations of the real face.

Lamé coefficients μ, λ are linear functions of Young's modulus $\mathbf{E}$ and Poission's ratio set to constant $\nu = 0.45$. The prestrained corotated term $\mathbf{P}_i(\vec{\mathbf{x}}, \vec{\mathbf{x}}_{\text{rest}}, \mathbf{G}_i) = \mathbf{R}_i\mathbf{F}_i(\vec{\mathbf{x}}, \vec{\mathbf{x}}_{\text{rest}})\mathbf{G}_i$ is obtained by factoring out rotational component of $\mathbf{F}_i\mathbf{G}_i$. The first term is enforcing the shape of the tetrahedron to match the muscle activation, the second term is minimizing relative volumetric change of the deformation. Passive tissues can be modeled by setting $\mathbf{D}_i$ to the 3×3 identity matrix. In practice, our tets are relatively coarse and may thus contain a mix of passive and active tissues, so $\mathbf{D}_i$ can be seen as their blending. We note that many different elasticity models could be used instead of corotated elasticity. Our choice was motivated mainly by tractability of the resulting inverse problem, as discussed later on.

Gravity. Accuracy of the gravity direction is one of the important factors in our material parameter estimation. Some 3D scanning devices (e.g., handheld scanners) do not provide information about a gravity vector and it is therefore necessary to optimize for it. We initialize a gravity potential $E_{\text{grav}}(\vec{\mathbf{x}}, \vec{\mathbf{g}})$ with a best guess of a gravity force vector $\vec{\mathbf{g}} \in \mathbb{R}^3$ corresponding to each of our surface scan.

Constraints. We use Dirichlet boundary conditions corresponding to the bones (the skull and the jaw), expressed as $\vec{\mathbf{c}}_{\text{bone}}(\vec{\mathbf{x}}, \mathbf{J}) = 0$, where $\mathbf{J} \in SE(3)$ is a time-varying rigid transformation of the jaw relative to the skull. The $\vec{\mathbf{c}}_{\text{bone}}(\vec{\mathbf{x}}, \mathbf{J})$ function fixes the vertices of the skull in place, while vertices of the jaw are rigidly transformed by $\mathbf{J}$.

Forward problem. With this model, quasi-static facial deformations can be computed by minimizing $E_{\text{elast}}(\vec{\mathbf{x}}, \alpha, \mathcal{H}) + E_{\text{grav}}(\vec{\mathbf{x}}, \vec{\mathbf{g}})$ subject to $\vec{\mathbf{c}}_{\text{bone}}(\vec{\mathbf{x}}, \mathbf{J}) = 0$. The inverse problem, i.e., computing both $\mathcal{H}$ (expression-independent parameters) as well as muscle activations α and jaw transformations $\mathbf{J}$ (expression-dependent parameters) which best explain our training 3D scans of facial expressions is explained in the following subsection.

5.4.3 Inverse problem formulation

The input data to our inverse modeling module are registered 3D scans of facial deformations and tetrahedral template mesh as shown in Figure 5.20.

We denote the vertices of the "target" 3D scans as $\vec{\mathbf{t}}_k \in \mathbb{R}^{3p}$, where k is the target scan index and p the number of vertices in the triangle mesh representing the skin (the same for all targets due to registration). Because our tet-mesh is volumetric and $\vec{\mathbf{x}} \in \mathbb{R}^{3n}$ thus includes internal vertices, we define a sparse selector matrix $\mathbf{S} \in \mathbb{R}^{3p \times 3n}$ such that $\mathbf{S}\vec{\mathbf{x}}$ "selects" the skin vertices corresponding to the targets $\vec{\mathbf{t}}_k$. In the inverse optimization problem we will also solve for the rest pose $\vec{\mathbf{x}}_{\text{rest}} \in \mathbb{R}^{3n}$, which is the configuration of vertices in static equilibrium subject to prestrains and bone constraints, but *not* subject to gravity. In theory, $\vec{\mathbf{x}}_{\text{rest}} \in \mathbb{R}^{3n}$ would be directly observable, e.g. during freefall, but in practice, we solve for $\vec{\mathbf{x}}_{\text{rest}}$ along with the other trainable parameters of our model. Our "inverse physics" can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\vec{\mathbf{x}}_{\text{rest}}, \vec{\mathbf{x}}_k, \alpha_k, \mathbf{J}_k, \vec{\mathbf{g}}_k, \mathcal{H}} \quad & \|\mathbf{S}\vec{\mathbf{x}}_k - \mathbf{T}_k\vec{\mathbf{t}}_k\|^2 + E_{\text{reg}}(\alpha_k, \vec{\mathbf{x}}_{\text{rest}}, \mu, \mathbf{G}, \mathbf{D}) \\ \text{subj. to} \quad & \nabla E_{\text{elast}}(\vec{\mathbf{x}}_{\text{rest}}, \vec{\mathbf{x}}_k, \alpha_k, \mathcal{H}) + \nabla E_{\text{grav}}(\vec{\mathbf{x}}_k, \vec{\mathbf{g}}_k) = 0 \\ & \vec{\mathbf{c}}_{\text{bone}}(\vec{\mathbf{x}}_{\text{rest}}, \mathbf{I}) = 0, \vec{\mathbf{c}}_{\text{bone}}(\vec{\mathbf{x}}_k, \mathbf{J}_k) = 0, \vec{\mathbf{c}}_{\text{col}}(\vec{\mathbf{x}}_k) \geq 0 \end{aligned} \tag{5.7}$$

This problem simultaneously solves for multiple tet-mesh deformations $\vec{\mathbf{x}}_k \in \mathbb{R}^{3n}$, where k indexes target 3D scans. In the objective function $\|\mathbf{S}\vec{\mathbf{x}}_k - \mathbf{T}_k\vec{\mathbf{t}}_k\|^2$, the $\mathbf{T}_k \in SE(3)$ is a rigid transformation applied to the target triangle mesh $\vec{\mathbf{t}}_k$ in order to compensate for global rigid motions of the skull (a process known as "rigid stabilization" Beeler and Bradley [2014]). E_{reg} is a regularization term discussed in the Section 5.4.4. The constraint $\vec{\mathbf{c}}_{\text{bone}}(\vec{\mathbf{x}}_{\text{rest}}, \mathbf{I}) = 0$ means that the jaw in the rest pose is closed (its rigid transformation is identity). In contrast, in non-neutral expressions the jaw vertices are rigidly transformed by $\mathbf{J}_k \in SE(3)$. The constraints $\vec{\mathbf{c}}_{\text{col}}(\vec{\mathbf{x}}_k) \geq 0$ are used to resolve collisions: if we find a vertex in $\vec{\mathbf{x}}_k$ penetrating a tetrahedron, we find projection onto the closest surface triangle and add a half-space constraint to $\vec{\mathbf{c}}_{\text{col}}(\vec{\mathbf{x}}_k) \geq 0$ which pushes the vertex out of the collision. Finally, the constraint $\nabla E_{\text{elast}}(\vec{\mathbf{x}}_{\text{rest}}, \vec{\mathbf{x}}_k, \alpha_k, \mathcal{H}) + \nabla E_{\text{grav}}(\vec{\mathbf{x}}_k, \vec{\mathbf{g}}_k) = 0$ requests the deformed pose $\vec{\mathbf{x}}_k$ to be in quasi-static equilibrium subject to gravity and boundary conditions. Our target 3D scans intentionally contain different head orientations, i.e., the gravity directions for each target can be different and are represented by trainable parameter $\vec{\mathbf{g}}_k \in \mathbb{R}^3$.

5.4.4 Solving the inverse problem

Solving Eq. 5.7 numerically is challenging. Our optimization problem has many variables because we are solving for deformed vertex positions $\vec{\mathbf{x}}_1 \in \mathbb{R}^{3n}, \vec{\mathbf{x}}_2 \in \mathbb{R}^{3n}, \ldots$ for all of our targets. Furthermore, the deformed vertex positions are coupled through the rest pose $\vec{\mathbf{x}}_{\text{rest}}$ and shared mechanical parameters $\mathbf{E}, \mathbf{G}, \mathbf{D}$, which makes the problem *not* separable per each target. To further complicate matters, the problem is ill-posed, because our data $\vec{\mathbf{t}}_k$ correspond only to skin (surface) measurements, but we are solving for unknown volumetric properties.

Our optimization strategy is to use the idea of homotopy. In our case, we use a progression from an initial state $\vec{\mathbf{x}}_{\text{init}}$ which is in a quasi-static equilibrium, but does not match target scans, to a state $\vec{\mathbf{x}}$ fitting all target scans as close as possible in a sense of Eq. 5.7. Even though asymptotic numerical method Chen et al. [2014] can provide better convergence than standard Newton-type optimization, the implementation for a general deformation problem is challenging. We found that a linear increase of the target term weight (from 0 to 1), iterative update of regularization terms E_{reg} and block coordinate descent optimization proved to be a good strategy for stable convergence and allowed us to prepare an automatic procedure for the complete inverse optimization.

Gravity direction initialization. In the first phase, we fit a set of six scans with neutral expression but with varying directions of gravity (see the top row in Figure 5.23) by fitting a passive model, i.e., we set $\mathbf{D} = \mathbf{I}$ and $\alpha = 0$. We compute an initial guess of the gravity-free rest pose $\vec{\mathbf{x}}_{\text{rest}}^{\text{init}}$ by averaging all of our 3D scans of the neutral expression under different directions of gravity. This is a good initial guess because for each gravity direction $\vec{\mathbf{d}}$, our dataset also contains the opposite direction $-\vec{\mathbf{d}}$, e.g., for a supine head orientation (facing upwards), we also have a prone orientation (facing downwards). We also use the $\vec{\mathbf{x}}_{\text{rest}}^{\text{init}}$ as a regularization term for $\vec{\mathbf{x}}_{\text{rest}}$.

Block coordinate descent. Next, we set $\mathbf{G} = \mathbf{I}$ and optimize for the heterogenous material properties $\mathbf{E}$. We assume constant density of soft tissues. In the case of quasi-statics, variation of densities and stiffness results in ambiguous fitting. If an MRI scanner is available, some sequences may provide a better initialization of densities. Optimizing for spatially varying material parameters has been recognized as a challenge in previous work Wang et al. [2015], Kim et al. [2017], which proposed to simplify the problem by sampling material properties only in a sparse set of points (such as 130 points). This reduces the number of variables, but also reduces expressivity and relies on well-chosen locations of these sample points, which is non-trivial in the face. Instead, we solve for a separate $\mathbf{E}$ for each tet by progressive refinement. The process starts by optimizing a single $\mathbf{E}$ for all tets (homogenous material), which theoretically corresponds to an infinitely strong Laplacian regularization term. Specifically, we use the standard umbrella operator penalizing $\|\mathbf{E}_a - \mathbf{E}_b\|^2$ summed over all pairs (a, b) of adjacent tets, as in Bickel et al. [2009]. Next, we use the homogeneous $\mathbf{E}$ as an initial guess and progressively reduce the weight of the Laplacian regularization term. This way, we find heterogeneous stiffness parameters E but so far only with isotropic material. Next, we allow the optimizer to optimize for prestrains $\mathbf{G}$ (along with $\mathbf{E}$ and $\vec{\mathbf{x}}_{\text{rest}}$), introducing anisotropy into our model. Finally, we enable optimization of muscle parameters: the time-independent maximal muscle deformations $\mathbf{D}$ and the corresponding time-dependent activations α.

Regularization. We applied additional regularization on $\mathbf{E}$ and $\mathbf{G}$ with a closeness term to an initial guess which we update in each iteration k: $\|\mathbf{G}_i^k - \mathbf{G}_i^{k-1}\|_F^2$ (regularization plasticity). Finally, we enable optimization of the muscle parameters $\mathbf{D}, \alpha$. We use an order of magnitude stronger weight for the muscle activation regularization $\|\mathcal{A}(\alpha_i, \mathbf{D}_i) - \mathbf{I}\|_F^2$ to enforce principle of least action (Section 5.4.2).

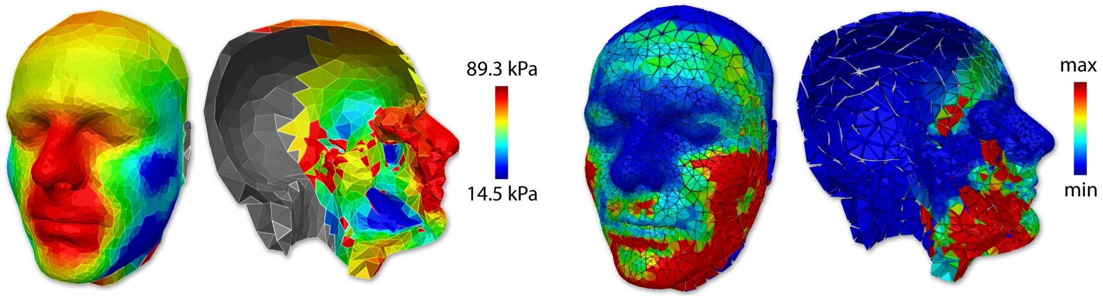


Figure 5.21: Visualization of heterogenous stiffness of our model and prestrain at each tetrahedron.

Jaw kinematics and stabilization. Additional free variables of our optimization process include kinematic parameters of the jaw ($\mathbf{J}_k$). We allow arbitrary rigid transformation, parameterized with three DoFs for translation and three Euler angles for rotation. We did not encounter any singularities with the Euler angles because the set of physiological rotations of the jaw is limited. We used the same parametrization for global head transformation (rigid stabilization) $\mathbf{T}_k$, taking advantage of our "pre-stabilization" to ensure that changes of global orientation at solver time will be small. Similarly, we parameterize gravity directions with two angles (polar and azimuthal angle, the magnitude is fixed to $\|\vec{\mathbf{g}}_k\| = 9.81m/s^2$).

Implementation details. We solve our optimization problems using an open source interior point solver IPOPT Wächter and Biegler [2006] with PARDISO Schenk and Gärtner [2004] as the underlying linear solver. Our mechanical model is twice differentiable. In most cases, we provide the solver with exact gradient and Hessian. In the inverse problem, the quasi-static constraint Hessian is approximated. An exact Hessian contains third derivative of the mechanical model resulting from the chain rule which we ignore Bickel et al. [2012]. Our problem converges in total of 25 iterations (5 iterations per each phase) and takes approximately 20 minutes for the highest resolution tetmesh (21k tetrahedra) and 13 targets scans on a desktop CPU.

5.4.5 Results

Stiffness. We visualize our resulting stiffness parameter in Figure 5.21. The computed Young's modulus values are in range of $\mathbf{E} \in [14.5, 89.3]$ with an average of $\mathbf{E} = 68.2kPa$. These values are aligned with in-vivo stiffness measurements Luboz et al. [2014], Shinohara et al. [2010]. However, material properties reported in the literature must be interpreted judiciously, e.g., Lapeer et al. [Lapeer et al., 2011] surveyed literature on skin stiffness measurements and found large differences between the reported values (attributed to differences between in-vivo/in-vitro measurements, nonlinearity of the skin and differences in elastic models). In Figure 5.21 we can see that the stiffness is lowest in areas such as the cheeks which contain the thickest layers of adipose tissues. Note that stiffness in the lips region seem to be too large. We observed that in case of large noise in the data (e.g., lip motion fitted by muscle deformation apparent in Figure 5.23), our isotropic stiffness fitting failed to capture this deformation as passive. This was not the case for experiments using only selected scans with minimal lip motion. Fortunately,

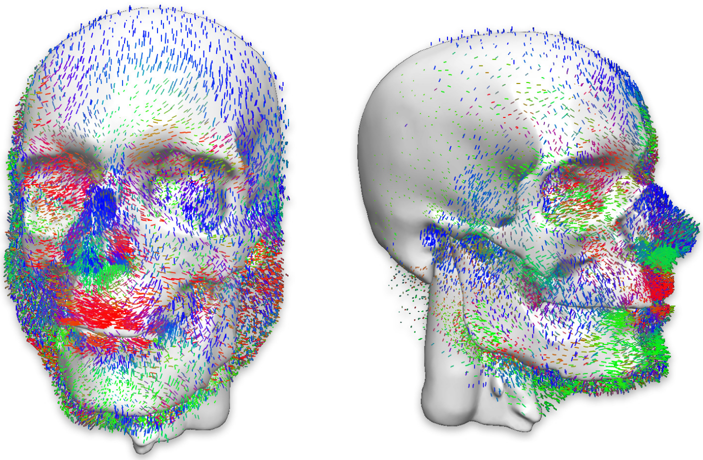


Figure 5.22: Visualization of principal muscle fiber directions (color coding: red = x, green = y, blue = z).

the anisotropic prestrain model is more flexible and is able to generalize better as shown in Figure 5.23.

Prestrain. Prestrain models the fact that soft tissues retract when excised from the body. This means that each tet in our mesh has a relaxed configuration with zero prestrain. Visualization of such configuration becomes more complicated because tet vertices corresponding to neighboring tets might not be aligned. Nevertheless, we can visualize prestrains by disconnecting our tet-mesh into isolated tets, scaling them down by a factor of 0.8, and transforming them by $\mathbf{G}$, see Figure 5.21. The color corresponds to prestrain magnitude computed as $\|\mathbf{G}-\mathbf{I}\|_F^2 \in [0, 3.7]$.

Muscle activations. Recall that our $\mathbf{D}$ matrices specify the maximal deformation of each tet due to muscle contractions. The $\mathbf{D}$ matrices are the same for all facial expressions, only the activation parameters α are allowed to vary from one expression to another. To visualize our resulting $\mathbf{D}$ matrices, we first apply polar decomposition and find the rotational parts are almost exactly identities. This is because rotations are already present in prestrain. Because $\mathbf{D}$ are very close to symmetric matrices, we can visualize their principal eigenvectors, which can be interpreted as muscle fiber directions (directions of maximal muscle contraction). In Figure 5.22, we visualize these principal eigenvectors as small line segments with the usual RGB color coding of directions (red = x, green = y, blue = z). We note that the resolution of our finest tet-mesh is still coarse relative to the shape of thin facial muscles and therefore these directions should be interpreted as weighted averages over all facial muscles intersecting a given tet.

Evaluation of passive model. Validation of the passive model is challenging due to residual muscle activation which is present in all facial expressions of healthy human beings. While measuring EMG signal is technically possible, each sensor measures electrical activity integrated over multiple muscles at the surface

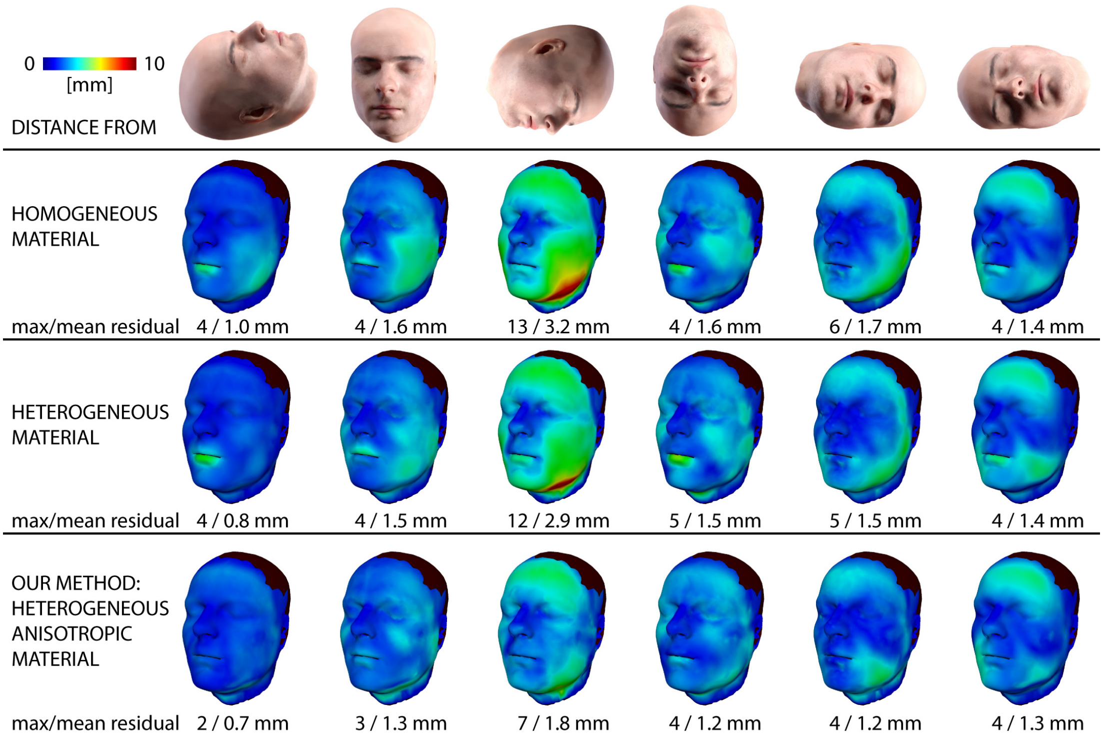


Figure 5.23: Comparison of elastic models – distances from 3D scans in various gravity directions (individually excluded from training data) shown in the first row (reference measurements). Second row: simulation with homogeneous material; Third row: heterogenous material without prestrain; last row: heterogeneous material with anisotropic prestrain (our method).

and the electrode placement affects the deformation. Ideal validation data would contain surface scans of a completely paralyzed face under different external loads.

In Figure 5.23 we analyze how well our model predicts 3D scans under varying gravity directions. The largest difference of 14mm is observed with prone head orientation (facing downwards), where the soft tissues sag the most under gravity. In the first row, we compare against simulated homogeneous material (single μ for all tets), which is the elasticity model used in Section 5.3. Methods using elastic model with homogeneous material (e.g.,Mollemans et al. [2007], Ichim et al. [2017]) either can not fit similar scans properly or need to compensate by artificial muscle deformation. We can see that homogeneous elasticity does not accurately predict deformations of the face due to gravity. In the second row, we show the result of our method *without* prestrain, i.e., setting $\mathbf{G} = \mathbf{I}$. In the third row is the final result of our method with prestrain enabled, achieving the most accurate prediction of the 3D scans.

Validation of a complete mechanical model. The parameters of our model were computed from a total of 13 3D scans constituting our training dataset. An important question is how closely can our model predict unseen *validation* 3D scans. To quantify this, we captured and registered three new 3D scans with different facial expressions, two of them in the supine orientation and one sitting. To fit these new expressions with our model, we fixed the human-specific parameters $\mathcal{H}$ which are time invariant and allowed our optimizer to modify only the time-varying muscle activations α and the rigid jaw transformation $\mathbf{J}$, which correspond to properties the human subject can voluntarily control to achieve different facial expressions. This worked quite well with our final model.

In Figure 5.24, we can see that the distances between predictions from our model and the actual 3D scans are similar for both training and validation 3D scans, i.e., overfitting is not an issue. We can also see that the effect of gravity is correctly captured by passive elasticity, unlike previous face animation models which explain all deformations by (overly) generalized muscle activations Ichim et al. [2016, 2017] and Section 5.3.

We were also quite excited about the fact the time-varying muscle activations α, shown in the second row of Figure 5.24, are localized even though we did not introduce any regularization terms encouraging locality; even more importantly, the activation signals seem to correspond to actual anatomical muscle groups responsible for creating facial expressions such as raising one corner of the mouth or pulling the lips forward (pucker). We can also notice that activations α corresponding to a partial smile are not just an interpolation of a full smile; more pronounced expressions are generated by engaging additional muscle groups.

Input data ambiguity. We test our method with different subsets of input scans to show the effect of ambiguity. Figure 5.25 shows optimized stiffness parameters for subsets **a-e**. The subset **a** uses only one scan. Since we're also optimizing for the rest-pose, the result mostly depends on regularization weights. Subsets **b** and **c** use different subset of size two - showing different results depending on how close input scans are with respect to gravity direction. Subset **d** with three scans shows a result similar to a set using all six gravity directions.

Template skull adaptation. We tested our method on a less accurate skull and mandible geometry adapted from a template model to assess the importance of precise segmentation of bones from MRI data. We started by smoothing high-frequency details on both template and target model. Next, we registered both models with manually picked keypoints and prepared a tetrahedral mesh that corresponds to the original surface mesh and the adapted skull. The result of our inverse problem in Figure 5.26 shows that stiffness parameters near the surface are similar and only small changes near areas of thin soft structures are present. Please note that the template skull geometry after the adaptation is not very different from the segmented skull (maximum distance is 16mm). Adapting skull to different subjects with varying fat distribution is a challenging task Achenbach et al. [2018].

Please refer to the accompanying video A.3 to see more detailed results.

5.4.6 Limitations and Future Work

Even though we believe our project is a significant step towards realistic facial modeling, there are several limitations leading to opportunities for future work.

Input data. We tested our method with only one human subject. Our 3D scanning process also needs to be improved, because our structured-light scanner

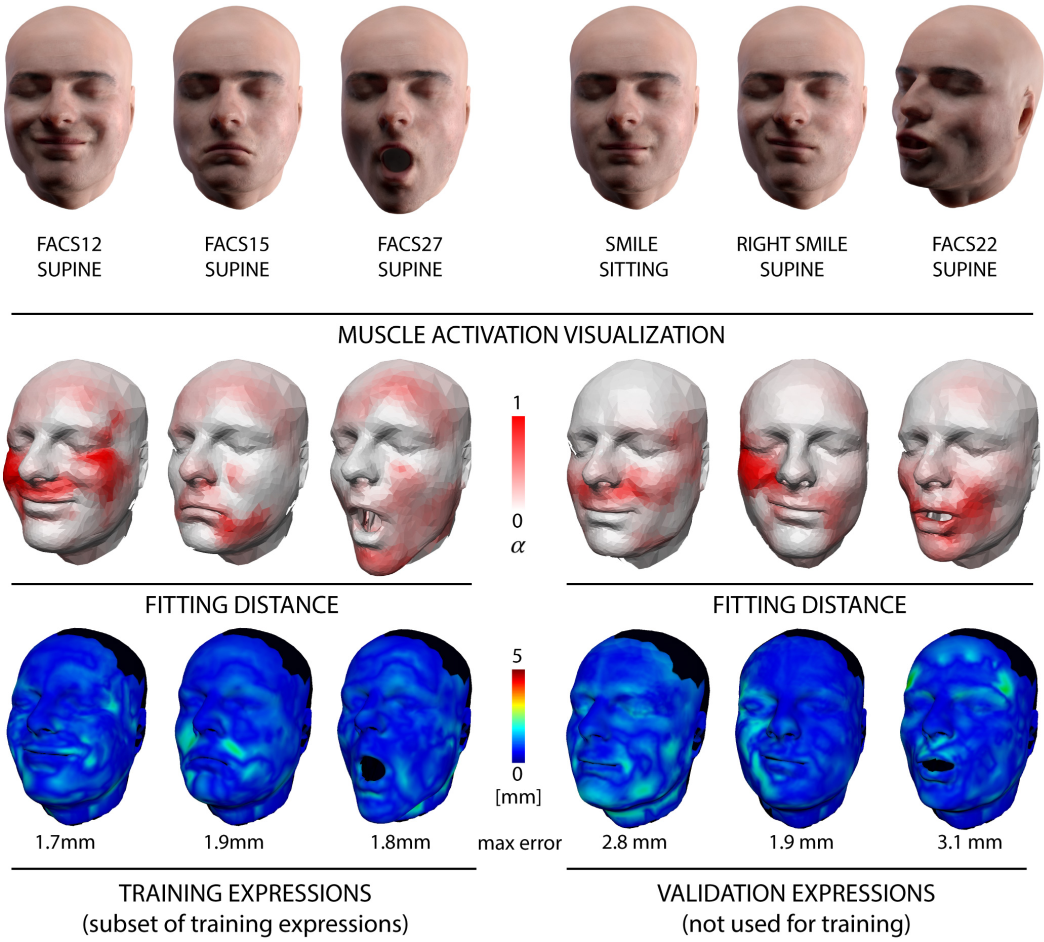


Figure 5.24: Explaining various facial expressions with our model using a fixed set of muscle deformations **D** visualized in Figure 5.22. Second row: visualization of our time-varying muscle activation parameters α used to generate the corresponding facial expressions with our method. Last row: testing the generalization capability of our model by comparing fitting errors on training (left three columns) and validation data (right three columns).

(Artec Spider) is not ideal for capturing facial expressions. One scan can take up to several minutes which makes scanning challenging due to muscle fatigue.

Registration. Incorrect registration (e.g., around lips) and rigid stabilization is causing some errors in the fitting. Ideally, our algorithm should run the inverse problem together with corrections for inaccurate correspondences.

Mechanical model. Our current mechanical model also leaves room for improvement. We note that many different constitutive equations could be used instead of corotated elasticity. Our choice was motivated mainly by tractability of the resulting inverse problem. The additional prestrain can be understand as a simple way to improve an arbitrary elastic model. Whether the anisotropic behavior would generalize well under larger loads is subject to further testing.

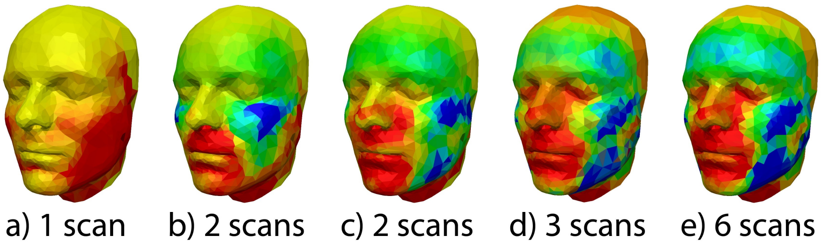


Figure 5.25: Stiffness parameters computed from limited scan data.

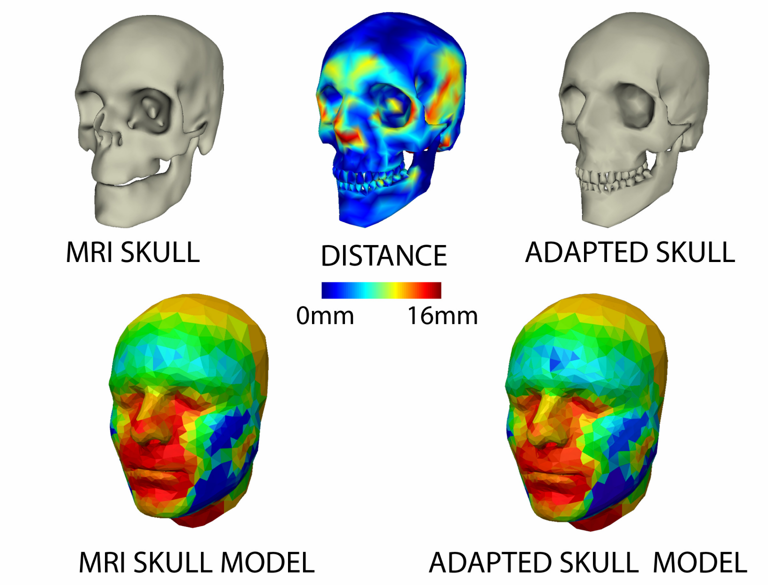


Figure 5.26: The first row shows our original skull segmented from MRI scan, the adapted skull from a template model and color-coded distance. The second row shows stiffness computed using the original model and the model with the adapted skull.

Muscle control. We briefly experimented with a model without kinematic jaw constraints where the motion of the jaw was driven purely by muscle activation. The model was able to fit all of our scans with jaw open and we are planning to experiment with purely muscle-driven control without kinematic constraints in the future.

Surface scan ambiguity. The current parameter learning is limited by surface scan observations. Simple synthetic experiments can show that multiple material configurations can lead to almost identical surface deformation under various conditions. Different ways of applying and measuring external loads should be therefore studied. Note that aspiration device testing might not reveal deep non-linearity without damaging the skin Luboz et al. [2014].

Dynamics. Our current physics-based modeling methodology assumes quasi-static deformations corresponding to slowly moving faces. Even though we can of course add ad-hoc dynamics effects, the proper approach would involve learning viscoelastic properties from data – dynamic deformations of real faces, which would require a dynamic (4D) capture studio, as was recently explored in the case of human bodies Kim et al. [2017]. In this project we focused on volumetric modeling of facial soft tissues; production of realistic facial animations would require adding accurate models of hair, teeth, tongue, eyes and eyelids.

Conclusion

Summary

In this dissertation, we presented methods for automated reconstruction of animatable virtual humans from sparse surface data. The theme of this work was based on our belief that by leveraging existing knowledge in human anatomy and physics, we can achieve better reconstruction with less data and provide semantically meaningful control of our models.

Human body modeling. In chapter 4, we presented an automatic method to reconstruct an anatomical, physics-based model of the body of a given human subject. To our knowledge, our system is the first to reconstruct personalized fully volumetric physics-based human body models, which are suitable for computer animation including effects such as inertia, collisions, and gravity. We believe that our method will serve as a tool for reducing the costs of person-specific modeling, and may inspire applications even beyond the traditional realms of computer graphics. Please refer to Section 4.10 for more detailed discussion on limitations.

Human face modeling. In chapter 5, we explored two models of the human face. Section 5.3 describes a physics-based simulation approach to face animation that complements existing generative methods such as blendshapes. These purely geometric methods can produce artifacts such as self-intersections in facial poses that were not specifically considered during the modeling of the blendshape basis – ensuring consistency in all possible linear combinations quickly becomes intractable. Even more challenging is the correct handling of dynamic effects such as interactions with external objects or inertial deformations. Please refer to Section 5.3.7 for more detailed discussion on limitations.

In Section 5.4, we explored more biomechanically accurate model. Passive soft tissue deformation may initially seem less important than e.g. muscle activations. However, apart from differences in blood flow, the change in shape of the face can be quite surprising. The core idea presented in this section was to overcome the challenging problem of capturing mechanical properties of biological tissues in-vivo from limited data. We believe our approach is an interesting complement to machine learning models that typically require very large training data sets. Our method has been designed to minimize the amount of training data by taking advantage of domain-specific priors, such as the principles of soft tissue elasticity. These priors contain trainable parameters such as soft tissue material parameters, which we estimated from training data. Please refer to Section 5.4.6 for more detailed discussion on limitations.

We believe our work shows an interesting research path towards modeling and reconstruction of virtual humans that crosses boundaries of computer graphics, physics-based animation and biomechanics.

Challenges

Model generalization. An important factor we needed to take into account during our research was generalization of our models to a large variety of characters. Applying strict anatomical models and elastic systems can easily make the space of possible reconstructed characters too narrow which is usually considered a significant disatvantage of physics-based models compared to purely data-driven models in computer graphics. We tried to tackle this problem by exploring flexibility and adjusting existing biomechanical models to find a balance between overfitting and generalization, e.g., by adding additional degrees of freedom to overly strict models and carefully regularize to keep plausibility at a level suitable for computer animation.

Level of modeling detail. Another key decision was to choose an appropriate level of modeling detail needed for reconstruction targeting computer animation at the scale we can display on a monitor, TV or in VR. For example, humans are very suspectible to small deformations of a face due to contraction of rather small muscles in contrast to other skeletal muscles that we primarily use to move our body in space. This means that we needed to choose a different level of detail of modeling for a face and body and as was shown in our work, this does not simply mean adjusting resolution of a mesh but it requires a different formulation of the model. In general, our models can be considered large-scale compared to typical models in biomechanics where researchers focus on a part of a human body at a smaller scale (e.g., Tran et al. [2007]). In our work, we did not implement such accurate models but rather focused on the overall visual plausibility.

Future work

We only focused on the most prominent parts of the human body. To achieve realistic reconstruction, modeling additional parts of the body such as hair or teeth is important.

Thoroughout all of presented models, we apply dynamics only in the forward modeling and ignore inertial effects in input data. Integrating dynamics in the inverse problem would not only reduce issues with input data, but would also give us a possibility to optimize properties of the dynamical system such as damping.

Constitutive models we used were still rather a crude approximation of a realistic behavior observed on soft tissues. Better passive and active models and more complex layering of materials would definitely improve believability.

One of the biggest challenge when working with complex computational models such as ours is performance. The resulting detail of the reconstructed mesh is lower compared to state of the art in rendering. Improving performance of a physics-based animation systems is still an open problem in computer animation. In our case, the inverse problem also introduces specific computational challenges in solving linear systems. Efficient formulation of the inverse problem also poses intriguing research questions, while advances in numerical optimization could help with solving the large and non-linear optimization problem.

Finally, if data availability would not be a concern, a combination of data-driven and physics-based modeling could help tackle many challenges such as parameterization and modeling of adipose tissue distribution or learning a constitutive model [Wang et al., 2018] of soft tissue layers from data.

Bibliography

Jascha Achenbach, Robert Brylka, Thomas Gietzen, Katja zum Hebel, Elmar Schömer, Ralf Schulze, Mario Botsch, and Ulrich Schwanecke. A multilinear model for bidirectional craniofacial reconstruction. In *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine*, pages 67–76. Eurographics Association, 2018.

Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The digital Emily project: Achieving a photorealistic digital actor. *Computer Graphics and Applications, IEEE*, 30 (4):20–31, 2010.

Jérémie Allard, Stéphane Cotin, Francois Faure, Pierre-Jean Bensoussan, Francois Poyer, Christian Duriez, Hervé Delingette, and Laurent Grisoni. Sofa-an open source framework for medical simulation. In *MMVR 15-Medicine Meets Virtual Reality*, volume 125, pages 13–18. IOP Press, 2007.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Trans. Graph.*, volume 24, pages 408–416, 2005.

Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. Inverse kinematics techniques in computer graphics: A survey. In *Computer Graphics Forum*, number 6, pages 35–58. Wiley Online Library, 2018.

Lesley N Baldwin, Keith Wachowicz, Steven D Thomas, Ryan Rivest, and B Gino Fallone. Characterization, prediction, and correction of geometric distortion in 3t mr images. *Medical Physics*, 34(2):388–399, 2007.

Michael Bao, Matthew Cong, Stéphane Grabli, and Ronald Fedkiw. High-quality face capture using anatomical muscles. *CoRR*, abs/1812.02836, 2018. URL `http://arxiv.org/abs/1812.02836`.

GG Barbarino, M Jabareen, J Trzewik, A Nkengne, G Stamatas, and E Mazza. Development and validation of a three-dimensional finite element model of the face. *Journal of biomechanical engineering*, 131(4):041006, 2009.

Vincent Barrielle, Nicolas Stoiber, and Cedric Cagniart. Blendforces, a dynamic framework for facial animation. *Comp. Graph. Forum*, 2016.

Millard F Beatty. Topics in finite elasticity: hyperelasticity of rubber, elastomers, and biological tissues—with examples. *Appl. Mech. Rev. Dec 1987*, 1987.

Thabo Beeler and Derek Bradley. Rigid stabilization of facial expressions. *ACM Trans. Graph.*, 33(4):44, 2014.

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. In *ACM Transactions on Graphics (ToG)*, volume 29, page 40. ACM, 2010.

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. In *ACM Transactions on Graphics (TOG)*, volume 30, page 75. ACM, 2011.

Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph.*, 28(3), 2009.

Bernd Bickel, Peter Kaufmann, Mélina Skouras, Bernhard Thomaszewski, Derek Bradley, Thabo the, Phil Jackson, Steve Marschner, Wojciech Matusik, and Markus Gross. Physical face cloning. *ACM Trans. Graph.*, 31(4):118, 2012.

Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of the 26th annual conf. on Comp. graph. and interactive techniques*, pages 187–194, 1999.

Silvia S Blemker and Scott L Delp. Three-dimensional representation of complex muscle architectures and geometries. *Annals of biomedical engineering*, 33(5): 661–673, 2005.

Silvia S Blemker, Peter M Pinsky, and Scott L Delp. A 3d model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of biomechanics*, 38(4):657–665, 2005.

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition*, 2014.

Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4):154, 2014.

Andrew M Bradley. Pde-constrained optimization and the adjoint method. Technical report, Technical Report. Stanford University. https://cs. stanford. edu/˜ ambrad . . . , 2013.

Vicki Bruce and Andy Young. Understanding face recognition. *British journal of psychology*, 77(3):305–327, 1986.

Grégory Chagnon, Marie Rebouah, and Denis Favier. Hyperelastic energy densities for soft biological tissues: a review. *Journal of Elasticity*, 120(2):129–160, 2015.

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph.*, 33(4):95, 2014.

Hon Fai Choi and Silvia S Blemker. Skeletal muscle fascicle arrangements can be reconstructed using a laplacian vector field simulation. *PloS one*, 8(10):e77576, 2013.

Matthew Cong, Michael Bao, Kiran S Bhat, Ronald Fedkiw, et al. Fully automatic generation of anatomical face simulation models. In *Proc. of the EG/SIGGRAPH Symposium on Comp. Anim.*, pages 175–183. ACM, 2015.

Matthew Cong, Kiran S Bhat, and Ronald Fedkiw. Art-directed muscle simulation for high-end facial animation. In *Proc. of the EG/SIGGRAPH Symposium on Comp. Anim.*, pages 119–127, 2016.

Bruce M Damon, Martijn Froeling, Amanda KW Buck, Jos Oudeman, Zhaohua Ding, Aart J Nederveen, Emily C Bush, and Gustav J Strijkers. Skeletal muscle diffusion tensor-mri fiber tracking: rationale, data acquisition and analysis methods, applications and future directions. *NMR in Biomedicine*, 30(3):e3563, 2017.

Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. Opensim: open-source software to create and analyze dynamic simulations of movement. *Biomedical Engineering, IEEE Transactions on*, 54(11):1940–1950, 2007.

Ali-Hamadi Dicko, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, Francois Faure, Olivier Palombi, and Marie-Paule Cani. Anatomy transfer. *ACM Trans. Graph.*, 32(6):188, 2013.

Renee Dunlop. *Production Pipeline Fundamentals for Film and Games.* Routledge, 2014.

Paul Ekman and Wallace V Friesen. Facial action coding system. 1977.

Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS).* Oxford University Press, USA, 1997.

Benjamin J Ellis, Trevor J Lujan, Michelle S Dalton, and Jeffrey A Weiss. Medial collateral ligament insertion site and contact forces in the acl-deficient knee. *Journal of Orthopaedic Research*, 24(4):800–810, 2006.

Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. A fast finite element solution for cloth modelling. In *11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings.*, pages 244–251. IEEE, 2003.

Remondino Fabio et al. From point cloud to surface: the modeling and visualization problem. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(5):W10, 2003.

Nele Famaey and Jos Vander Sloten. Soft tissue modelling for applications in virtual surgery and surgical robotics. *Computer methods in biomechanics and biomedical engineering*, 11(4):351–366, 2008.

Ye Fan, Joshua Litven, David IW Levin, and Dinesh K Pai. Eulerian-on-lagrangian simulation. *ACM Transactions on Graphics (TOG)*, 32(3):22, 2013.

Ye Fan, Joshua Litven, and Dinesh K Pai. Active volumetric musculoskeletal systems. *ACM Trans. Graph.*, 33(4):152, 2014.

Francois Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, et al. Sofa: A multi-model framework for interactive physical simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pages 283–321. Springer, 2012.

Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, et al. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic resonance imaging*, 30(9): 1323–1341, 2012.

Yuan-cheng Fung. *Biomechanics: mechanical properties of living tissues*. Springer Science & Business Media, 2013.

Babarenda Gamage, P Thiranja, Vijayaraghavan Rajagopal, Matthias Ehrgott, Martyn P Nash, and Poul MF Nielsen. Identification of mechanical properties of heterogeneous soft bodies using gravity loading. *International journal for numerical methods in biomedical engineering*, 27(3):391–407, 2011.

Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics*, 21(10):1103–1115, 2015.

Michael W Gee, Ch Förster, and WA Wall. A computational strategy for prestressing patient-specific biomechanical problems under finite deformation. *International Journal for Numerical Methods in Biomedical Engineering*, 26(1): 52–72, 2010.

Gael Guennebaud, Benoit Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

Lianghao Han, John H Hipwell, Christine Tanner, Zeike Taylor, Thomy Mertzanidou, Jorge Cardoso, Sebastien Ourselin, and David J Hawkes. Development of patient-specific biomechanical models for predicting large breast deformation. *Physics in Medicine and Biology*, 57(2):455, 2011.

Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A statistical model of human pose and body shape. In *Comp. Graph. Forum*, volume 28, pages 337–346. Wiley Online Library, 2009.

Hibbitt, Karlsson, and Sorensen. *ABAQUS/standard User's Manual*, volume 1. Hibbitt, Karlsson & Sorensen, 2001.

David A Hirshberg, Matthew Loper, Eric Rachlin, and Michael J Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *Computer Vision–ECCV 2012*, pages 242–255. Springer, 2012.

Alexandru Ichim, Ladislav Kavan, Merlin Nimier-David, and Mark Pauly. Building and animating user-specific volumetric face rigs. In *Proc. of the EG/SIGGRAPH Symposium on Comp. Anim.*, 2016.

Alexandru Eugen Ichim, the Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph.*, 2015.

Alexandru-Eugen Ichim, Petr Kadlecek, Ladislav Kavan, and Mark Pauly. Phace: Physics-based face modeling and animation. *ACM Trans. Graph.*, 36(4), 2017.

Geoffrey Irving, Joseph Teran, and Ron Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proc. of the EG/SIGGRAPH Symposium on Comp. Anim.*, pages 131–140, 2004.

Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78, 2011.

Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph.*, 32(4): 33, 2013.

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. Instant field-aligned meshes. *ACM Trans. Graph.*, 34(6):189:1–189:15, 2015. doi: 10.1145/2816795.2818078.

Petr Kadleček and Ladislav Kavan. Building accurate physics-based face models from data. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1–16, 2019.

Petr Kadlecek, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Krivanek, and Ladislav Kavan. Reconstructing personalized anatomical models for physics-based body animation. *ACM Trans. Graph.*, 35(6), 2016.

Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007.

Meekyoung Kim, Gerard Pons-Moll, Sergi Pujades, Seungbae Bang, Jinwook Kim, Michael J Black, and Sung-Hee Lee. Data-driven physics for human soft tissue animation. *ACM Transactions on Graphics (TOG)*, 36(4):54, 2017.

Yeara Kozlov, Derek Bradley, Moritz Bächer, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. Enriching facial blendshape rigs with physical simulation. In *Computer Graphics Forum*, volume 36, pages 75–84. Wiley Online Library, 2017.

Igor Kriz and Aleš Pultr. *Introduction to mathematical analysis.* Springer, 2013.

Lana Lan, Matthew Cong, and Ronald Fedkiw. Lessons from the evolution of an anatomical facial muscle model. In *Proceedings of the ACM SIGGRAPH Digital Production Symposium*, page 11. ACM, 2017.

Rudy J Lapeer, Paul D Gasson, and Vasudev Karri. A hyperelastic finite-element model of human skin for interactive real-time surgical simulation. *IEEE Transactions on Biomedical Engineering*, 58(4):1013–1022, 2011.

Dongwoon Lee, Michael Glueck, Azam Khan, Eugene Fiume, and Ken Jackson. A survey of modeling and simulation of skeletal muscle. *ACM Trans. Graph.*, 28(4):1–13, 2010.

Sung-Hee Lee and Demetri Terzopoulos. Heads up!: biomechanical modeling and neuromuscular control of the neck. In *ACM Trans. Graph.*, volume 25, pages 1188–1198, 2006.

Sung-Hee Lee and Demetri Terzopoulos. Spline joints for multibody dynamics. In *ACM Trans. Graph.*, volume 27, page 22, 2008.

Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph.*, 28 (4):99, 2009.

David IW Levin, Benjamin Gilles, Burkhard Mädler, and Dinesh K Pai. Extracting skeletal muscle fiber fields from noisy diffusion tensor data. *Medical Image Analysis*, 15(3):340–353, 2011.

John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of the 27th annual conf. on Comp. graph. and inter. tech.*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000.

John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. In *Eurographics (State of the Art Reports)*, pages 199–218, 2014.

Duo Li, Shinjiro Sueda, Debanga R Neog, and Dinesh K Pai. Thin skin elastodynamics. *ACM Trans. Graph.*, 32(4):49, 2013.

Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. In *ACM Trans. Graph.*, volume 29, page 32. ACM, 2010.

Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36(6):194:1–194:17, November 2017. Two first authors contributed equally.

Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Trans. Graph.*, 32(6):209:1–7, 2013.

Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. Quasi-newton methods for real-time simulation of hyperelastic materials. *Acm Transactions on Graphics (TOG)*, 36(3):1–16, 2017.

John E Lloyd, Ian Stavness, and Sidney Fels. Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In *Soft tissue biomechanical modeling for computer assisted surgery*, pages 355–394. Springer, 2012.

Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220, 2014.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

Vincent Luboz, Emmanuel Promayon, and Yohan Payan. Linear elastic properties of the facial soft tissues using an aspiration device: towards patient specific characterization. *Annals of biomedical engineering*, 42(11):2369–2378, 2014.

Wan-Chun Ma, Yi-Hua Wang, Graham Fyffe, Bing-Yu Chen, and Paul Debevec. A blendshape model that incorporates physical interaction. *Computer Animation and Virtual Worlds*, 23(3-4), 2012.

Steve A Maas, Benjamin J Ellis, Gerard A Ateshian, and Jeffrey A Weiss. Febio: finite elements for biomechanics. *Journal of biomechanical engineering*, 134(1): 011005, 2012.

Steve A Maas, Ahmet Erdemir, Jason P Halloran, and Jeffrey A Weiss. A general framework for application of prestrain to computational models of biological materials. *journal of the mechanical behavior of biomedical materials*, 61:499–510, 2016.

DC Martin, MK Medri, RS Chow, V Oxorn, RN Leekam, AM Agur, and NH McKee. Comparing human skeletal muscle architectural parameters of cadavers with in vivo ultrasonographic measurements. *The Journal of Anatomy*, 199(4): 429–434, 2001.

Aleka McAdams, Andrew Selle, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. Computing the singular value decomposition of 3x3 matrices with minimal branching and elementary floating point operations. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2011a.

Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. Efficient elasticity for character skinning with contact and collisions. In *ACM Trans. Graph.*, volume 30, page 37, 2011b.

L Angela Mihai, LiKang Chin, Paul A Janmey, and Alain Goriely. A comparison of hyperelastic constitutive models applicable to brain and fat tissues. *Journal of The Royal Society Interface*, 12(110):20150486, 2015.

Edward M Mikhail, James S Bethel, and J Chris McGlone. Introduction to modern photogrammetry. *New York*, 19, 2001.

Wouter Mollemans, Filip Schutyser, Nasser Nadjmi, Frederik Maes, and Paul Suetens. Predicting soft tissue deformations for a maxillofacial surgery planning system: from computational strategies to a complete clinical validation. *Medical image analysis*, 11(3):282–301, 2007.

Melvin Mooney. A theory of large elastic deformation. *Journal of applied physics*, 11(9):582–592, 1940.

Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36, 2014.

Matthias Müller and Markus H Gross. Interactive virtual materials. In *Graphics interface*, volume 4, pages 239–246, 2004.

Matthias Müller, Jos Stam, Doug James, and Nils Thürey. Real time physics: class notes. In *ACM SIGGRAPH 2008 classes*, pages 1–90. 2008.

Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation.* CRC press, 1994.

Mohammad Ali Nazari, Pascal Perrier, Matthieu Chabanas, and Yohan Payan. Simulation of dynamic orofacial movements using a constitutive law varying with muscle activation. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(4):469–482, 2010.

Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

Ray W Ogden. Nonlinear elasticity, anisotropy, material stability and residual stresses in soft tissue. In *Biomechanics of soft tissue in cardiovascular systems*, pages 65–108. Springer, 2003.

Ray W Ogden and Gerhard A Holzapfel. *Mechanics of biological tissue.* Springer, 2006.

Raymond W Ogden, Giuseppe Saccomandi, and Ivonne Sgura. Fitting hyperelastic models to experimental data. *Computational Mechanics*, 34(6):484–502, 2004.

Raymond William Ogden. Large deformation isotropic elasticity–on the correlation of theory and experiment for incompressible rubberlike solids. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 326(1567): 565–584, 1972.

U Ozsoy, R Sekerci, and E Ogut. Effect of sitting, standing, and supine body positions on facial soft tissue: Detailed 3d analysis. *International journal of oral and maxillofacial surgery*, 44(10):1309–1316, 2015.

Dinesh K Pai, Kees van den Doel, Doug L James, Jochen Lang, John E Lloyd, Joshua L Richmond, and Som H Yau. Scanning physical interaction behavior of 3d objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 87–96. ACM, 2001.

Frederick I Parke. Computer generated animation of faces. In *Proceedings of the ACM annual conference-Volume 1*, pages 451–457, 1972.

Joel E Pessa, Vikram P Zadoo, Peter A Garza, Erle K Adrian Jr, Adriane I Dewitt, and Jaime R Garza. Double or bifid zygomaticus major muscle: anatomy, incidence, and clinical correlation. *Clinical Anatomy: The Official Journal of the American Association of Clinical Anatomists and the British Association of Clinical Anatomists*, 11(5):310–313, 1998.

Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Trans. Graph.*, 34 (4):120, 2015.

Matthew D Robson and Graeme M Bydder. Clinical ultrashort echo time imaging of bone and other connective tissues. *NMR in Biomedicine*, 19(7):765–780, 2006.

MB Rubin and SR Bodner. A three-dimensional nonlinear model for dissipative response of soft tissue. *International Journal of Solids and Structures*, 39(19): 5081–5099, 2002.

Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

Shunsuke Saito, Zi-Ye Zhou, and Ladislav Kavan. Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans. Graph.*, 34(4), 2015.

Olaf Schenk and Klaus Gärtner. Solving unsymmetric sparse systems of linear equations with pardiso. *Future Generation Computer Systems*, 20(3):475–487, 2004.

Robert Schleip, Thomas W Findley, Leon Chaitow, and Peter Huijing. *Fascia: the tensional network of the human body: the science and clinical applications in manual and movement therapy.* Elsevier Health Sciences, 2013.

Ryan Schmidt and Karan Singh. Meshmixer: an interface for rapid mesh composition. In *ACM SIGGRAPH 2010 Talks*, page 6. ACM, 2010.

D Shahmirzadi, HA Bruck, and AH Hsieh. Measurement of mechanical properties of soft tissues in vitro under controlled tissue hydration. *Experimental Mechanics*, 53(3):405–414, 2013.

Minoru Shinohara, Karim Sabra, Jean-Luc Gennisson, Mathias Fink, and Mickaél Tanter. Real-time visualization of muscle stiffness distribution with ultrasound shear wave imaging during muscle contraction. *Muscle & nerve*, 42(3):438–441, 2010.

Ken Shoemake and Tom Duff. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface*, volume 92, pages 258–264. Citeseer, 1992.

Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2):11, 2015.

Weiguang Si, Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. Realistic biomechanical simulation and control of human swimming. *ACM Trans. Graph.*, 34(1):10, 2014.

Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, page 20, 2012.

Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM Trans. Graph.*, volume 24, pages 417–425, 2005.

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. Designing inflatable structures. *ACM Trans. Graph.*, 33(4):63, 2014.

Alex Smith, Sven Pohle, Wan-Chun Ma, Chongyang Ma, Xian-Chun Wu, Yanbing Chen, Etienne Danvoye, Jorge Jimenez, Sanjit Patel, Mike Sanders, and Cyrus A. Wilson. Emotion challenge: Building a new photoreal facial performance pipeline for games. In *Proceedings of the ACM SIGGRAPH Digital Production Symposium*, DigiPro '17, pages 8:1–8:2, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5102-7. doi: 10.1145/3105692.3105695. URL `http://doi.acm.org/10.1145/3105692.3105695`.

Breannan Smith, Fernando De Goes, and Theodore Kim. Stable neo-hookean flesh simulation. *ACM Transactions on Graphics (TOG)*, 37(2):1–15, 2018.

Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, 2007.

Ian Stavness, Mohammad Ali Nazari, Cormac Flynn, Pascal Perrier, Yohan Payan, John E Lloyd, and Sidney Fels. Coupled biomechanical modeling of the face, jaw, skull, tongue, and hyoid bone. In *3D Multiscale Physiological Human*, pages 253–274. Springer, 2014.

Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. In *ACM Trans. Graph.*, volume 23, pages 399–405, 2004.

TEN 24. 3d scan store. http://www.3dscanstore.com, 2016.

Joseph Teran, Sylvia Blemker, V Hing, and Ronald Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Proc. of the EG/SIGGRAPH Symposium on Comp. Anim.*, pages 68–74. Eurographics Association, 2003.

Joseph Teran, Eftychios Sifakis, Silvia S Blemker, Victor Ng-Thow-Hing, Cynthia Lau, and Ronald Fedkiw. Creating and simulating skeletal muscle from the visible human data set. *Visualization and Computer Graphics, IEEE Transactions on*, 11(3):317–328, 2005a.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. Robust quasistatic finite elements and flesh simulation. In *Proc. of the EG/SIGGRAPH Symposium on Comp. Anim.* ACM, 2005b.

Demetri Terzopoulos and Keith Waters. Physically-based facial modelling, analysis, and animation. *Computer Animation and Virtual Worlds*, 1(2):73–80, 1990.

The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.8 edition, 2016. URL `http://doc.cgal.org/4.8/Manual/packages.html`.

Angel Torrado-Carvajal, Joaquin L Herraiz, Eduardo Alcain, Antonio S Montemayor, Lina Garcia-Cañamaque, Juan A Hernandez-Tamames, Yves Rozenholc, and Norberto Malpica. Fast patch-based pseudo-ct synthesis from t1-weighted mr images for pet/mr attenuation correction in brain studies. *Journal of Nuclear Medicine*, 57(1):136–143, 2016.

HV Tran, F Charleux, M Rachik, Alain Ehrlacher, and MC Ho Ba Tho. In vivo characterization of the mechanical properties of human skin derived from mri and indentation techniques. *Computer methods in biomechanics and biomedical engineering*, 10(6):401–407, 2007.

Aggeliki Tsoli, Naureen Mahmood, and Michael J Black. Breathing life into shape: capturing, modeling and animating 3d human breathing. *ACM Trans. Graph.*, 33(4):52, 2014.

Christopher D Twigg and Zoran Kacic-Alesic. Point cloud glue: Constraining simulations using the procrustes transform. In *Symposium on Computer Animation*, pages 45–53, 2010.

Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002.

Javier von der Pahlen, Jorge Jimenez, Etienne Danvoye, Paul Debevec, Graham Fyffe, and Oleg Alexander. Digital ira and beyond: Creating real-time photoreal digital actors. In *ACM SIGGRAPH 2014 Courses*, SIGGRAPH '14, 2014. ISBN 978-1-4503-2962-0. doi: 10.1145/2614028.2615407. URL `http://doi.acm.org/10.1145/2614028.2615407`.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 2006.

Bridget M Waller, James J Cray Jr, and Anne M Burrows. Selection for universal facial emotion. *Emotion*, 8(3):435, 2008.

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. Deformation capture and modeling of soft objects. *ACM Trans. Graph.*, 34(4): 94, 2015.

Bin Wang, Paul Kry, Yuanmin Deng, Uri Ascher, Hui Huang, and Baoquan Chen. Neural material: Learning elastic constitutive material and damping models from sparse data. *arXiv preprint arXiv:1808.04931*, 2018.

Huamin Wang and Yin Yang. Descent methods for elastic body simulation on the gpu. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM Trans. Graph.*, volume 30, page 77. ACM, 2011.

Jeffrey A Weiss, Bradley N Maker, and Sanjay Govindjee. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer methods in applied mechanics and engineering*, 135(1):107–128, 1996.

WETA digital. Tissue system. http://www.fxguide.com/fxguidetv/fxguidetv-166-weta-digitals-tissue-system/, 2013.

Ge Wu, Sorin Siegler, Paul Allard, Chris Kirtley, Alberto Leardini, Dieter Rosenbaum, Mike Whittle, Darryl D Dâ€™Lima, Luca Cristofolini, Hartmut Witte, et al. Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motionâ€”part i: ankle, hip, and spine. *Journal of biomechanics*, 35(4):543–548, 2002.

Tim Wu. *A computational framework for modelling the biomechanics of human facial expressions.* PhD thesis, ResearchSpace@ Auckland, 2013.

Wenwu Yang, Nathan Marshak, Daniel Sýkora, Srikumar Ramalingam, and Ladislav Kavan. Building anatomically realistic jaw kinematics model from data. *CoRR*, abs/1805.05903, 2018. URL `http://arxiv.org/abs/1805.05903`.

Seung-Hyun Yoon, John Lewis, and Taehyun Rhee. Blending face details: Synthesizing a face using multiscale face models. *IEEE computer graphics and applications*, 37(6):65–75, 2017.

Eduard Zell, JP Lewis, Junyong Noh, Mario Botsch, et al. Facial retargeting with automatic range of motion alignment. *ACM Transactions on Graphics (TOG)*, 36(4):154, 2017.

Shuo Zhang, Arun A Joseph, Dirk Voit, Sebastian Schaetz, Klaus-Dietmar Merboldt, Christina Unterberg-Buchwald, Anja Hennemuth, Joachim Lotz, and Jens Frahm. Real-time magnetic resonance imaging of cardiac function and flow—recent progress. *Quantitative imaging in medicine and surgery*, 4(5):313, 2014.

Lifeng Zhu, Xiaoyan Hu, and Ladislav Kavan. Adaptable anatomical models for realistic bone motion reconstruction. *Comp. Graph. Forum*, 34(2), 2015.

Afra Zomorodian and Herbert Edelsbrunner. Fast software for box intersections. In *Computer Graphics Forum (Proc. of the EG/SIGGRAPH Symposium on Geom. Proc.)*, pages 129–138. ACM, 2000.

Gaspard Zoss, Derek Bradley, Pascal Bérard, and Thabo Beeler. An empirical rig for jaw animation. *ACM Transactions on Graphics (TOG)*, 37(4):59, 2018.

Silvia Zuffi and Michael J Black. The stitched puppet: A graphical model of 3d human shape and pose. In *Computer Vision and Pattern Recognition*, pages 3537–3546, 2015.

Zygote. Zygote body, 2016. URL `https://zygotebody.com`. [Online; accessed 28-Dec-2016].

List of Figures

List of Abbreviations

ARAP As-rigid-as-possible

CT Computerized tomography

DTI Diffusion tensor imaging

FEM Finite Element Method

FK Forward Kinematics

ICP Iterative closest point

IK Inverse Kinematics

MRI Magnetic resonance imaging

PBD Position-based Dynamics

PD Projective Dynamics

List of publications

This dissertation is based on the following publications:

Petr Kadleček, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Křivánek, and Ladislav Kavan. Reconstructing personalized anatomical models for physics-based body animation. ACM Transactions on Graphics, 35(6), 2016.

Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. Phace: Physics-based face modeling and animation. ACM Transactions on Graphics, 36(4), 2017.

Petr Kadleček and Ladislav Kavan. Building accurate physics-based face models from data. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2(2):1-16, 2019.

A. Attachments

A.1 Video 1: Inverse Human Body Modeling

This attachment is an accompanying video to the Section 4 and is available in an electronic form as a file:

- video1.mp4

Available online at `http://kadl.cz/petr/phd/video1.mp4`.

A.2 Video 2: Inverse Human Face Modeling

This attachment is an accompanying video to the Section 5.3 and is available in an electronic form as a file:

- video2.mp4

Available online at `http://kadl.cz/petr/phd/video2.mp4`.

A.3 Video 3: Inverse Human Face Modeling

This attachment is an accompanying video to the Section 5.4 and is available in an electronic form as a file:

- video3.mp4

Available online at `http://kadl.cz/petr/phd/video3.mp4`.