Physics-based Reconstruction and Animation of Humans

Alexandru - Eugen Ichim

Thèse n. 7880 2017
présenté le 23 Juin 2017
à la Faculté Informatique et Communications
laboratoire d'Informatique Graphique et Géométrique
programme doctoral en Informatique et Communications
École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Alexandru - Eugen Ichim

acceptée sur proposition du jury:

Prof. Martin Jaggi, président du jury
Prof. Mark Pauly, directeur de thèse
Prof. Pascal Fua, rapporteur
Prof. Ladislav Kavan, rapporteur
Dr. Thabo Beeler, rapporteur

Lausanne, EPFL, 2017

Vrei să cunoști lumea? Privește-o de aproape. Vrei să-ți placă? Privește-o de departe.
You want to know the world? Look at it closely. You want to like it? Look at it from afar.
— Ion Luca Caragiale

To my family…

Acknowledgements

I would like to start by thanking my thesis director, Mark Pauly, without whose supervision my research would not have been possible. I appreciate his patience, all his advice. During this time he was a guide for me both for my research, as well as for my personal development.

Almost as a second advisor, I would also like to send all my gratitude to Ladislav Kavan. The long enjoyable technical discussions I had with him were invaluable for the progress of my research.

On the list of people that influenced my research the most must be Sofien Bouaziz. He helped me get started with my very first paper and his help was what pushed me off the line. Petr Kadlecek, although he was an ocean away from my desk, during our projects together, we were almost like officemates. His hard work and incredible composure made the last couple of projects possible.

I would have not published without my coauthors. I would like to thank Mark Pauly, Ladislav Kavan, Petr Kadlecek, Federico Tombari, Merlin Nimier-David, Tiantian Liu, Jaroslav Krivanek, Dirk Holz, Felix Achilles, and Peter Bertholet. I am thankful to my thesis committee for all their input: Martin Jaggi, Mark Pauly, Pascal Fua, Ladislav Kavan, and Thabo Beeler.

Just before starting my PhD, people that introduced me to the magic of the scientific pursuit are Thibaut Weise and Radu Rusu. Without them I would not have collected the excitement necessary to launch into PhD studies. They showed me that papers are challenging, but can be fun, they can have amazing applications that people outside academia can appreciate and admire.

During these busy times, I had patient friends that supported me with their presence and well-being: Christopher, Vlad, Melvin, Teo, Simona, and Liliana.

Last but not least, I would like to thank my life partner, Cristina, for sharing the fun times, as well as the lesser ones during this entire period. She was the daily dose of happiness that I needed to keep me going.

Lausanne, 23 June 2017 A.E. I.

Abstract

Creating digital representations of humans is of utmost importance for applications ranging from entertainment (video games, movies) to human-computer interaction and even psychiatrical treatments. What makes building credible digital doubles difficult is the fact that the human vision system is very sensitive to perceiving the complex expressivity and potential anomalies in body structures and motion.

This thesis will present several projects that tackle these problems from two different perspectives: lightweight acquisition and physics-based simulation. It starts by describing a complete pipeline that allows users to reconstruct fully rigged 3D facial avatars using video data coming from a handheld device (e.g., smartphone). The avatars use a novel two-scale representation composed of blendshapes and dynamic detail maps. They are constructed through an optimization that integrates feature tracking, optical flow, and shape from shading. Continuing along the lines of accessible acquisition systems, we discuss a framework for simultaneous tracking and modeling of articulated human bodies from RGB-D data. We show how L1 regularization can be used to extract semantic information for the body shapes.

In the second half of the thesis, we will deviate from using standard linear reconstruction and animation models, and rather focus on exploiting physics-based techniques that are able to incorporate complex phenomena such as dynamics, collision response and incompressibility of the materials. The first approach we propose assumes that each 3D scan of an actor records his body in a physical steady state and uses a process called inverse physics to extract a volumetric physics-ready anatomical model of him. By using biologically-inspired growth models for the bones, muscles and fat, our method can obtain realistic anatomical reconstructions that can be later on animated using external tracking data such as the one resulting from tracking motion capture markers. This is then extended to a novel physics-based approach for facial reconstruction and animation. We propose a novel facial reconstruction and animation model which simulates biomechanical muscle contractions in a volumetric face model in order to create the facial expressions seen in the input scans. We then show how this approach allows for new avenues of dynamic artistic control, simulation of corrective facial surgery, and interaction with external forces and objects.

Key words: scanning, registration, face reconstruction, body reconstruction, simulation, facial animation, physics-based animation, body animation, face modeling, body modeling

Résumé

La création de représentations numériques d'humains revêt une importance capitale pour les applications allant du divertissement (jeux vidéo, films) à l'interaction homme-ordinateur et même aux traitements psychiatriques. Ce qui rend difficile le renforcement des doubles numériques est le fait que le système de vision humaine est très sensible à la perception de l'expressivité et des anomalies potentielles dans les structures et le mouvement du corps.

Cette thèse présentera plusieurs projets qui abordent ces problèmes sous deux angles différents : l'acquisition légère et la simulation basée sur la physique. Il commence par décrire un pipeline complet qui permet aux utilisateurs de reconstruire des avatars faciaux 3D complètement gréés en utilisant des données vidéo provenant d'un périphérique de poche (par exemple, un smartphone). Les avatars utilisent une nouvelle représentation à deux niveaux composée de formes de fond et de cartes détaillées dynamiques. Ils sont construits grâce à une optimisation qui intègre le suivi des fonctionnalités, le flux optique et la forme à partir de l'ombrage. En suivant les systèmes d'acquisition accessibles, nous discutons d'un cadre pour le suivi simultané et la modélisation de corps humains articulés à partir de données RGB-D.

Au cours de la deuxième moitié de la thèse, nous allons nous éloigner de l'utilisation de modèles de reconstruction et d'animation linéaire standard et nous concentrons plutôt sur l'exploitation de techniques basées sur la physique capables d'intégrer des phénomènes complexes tels que la dynamique, la réponse aux collisions et l'incompétence des matériaux. La première approche que nous proposons suppose que chaque analyse 3D d'un acteur enregistre son corps dans un état physique stable et utilise un processus appelé physique inverse pour extraire un modèle anatomique volumétrique prêt à la physique. En utilisant des modèles de croissance biologiquement inspirés pour les os, les muscles et les matières grasses, notre méthode peut obtenir des reconstructions anatomiques réalistes qui peuvent être ultérieurement animées en utilisant des données de suivi externes telles que celles résultant du suivi des marqueurs de capture de mouvement. Ceci est ensuite étendu à une nouvelle approche basée sur la physique pour la reconstruction et l'animation du visage. Nous proposons un nouveau modèle d'animation faciale qui simule des contractions musculaires biomécaniques dans un modèle de visage volumétrique afin de créer les expressions faciales observées dans les scans d'entrée. Nous montrons ensuite comment cette approche permet de nouvelles avenues de contrôle artistique dynamique, la simulation de la chirurgie corrective du visage et l'interaction avec des forces et des objets externes.

Mot clefs : scanning, registration, face reconstruction, body reconstruction, simulation, facial animation, physics-based animation, body animation, face modeling, body modeling

Contents

Acknowledgements **i**

Abstract (English/Francais) **iii**

1 Introduction **1**

1.1 Motivation 1

1.2 Publications 2

1.3 Organization 3

List of figures **1**

List of tables **1**

2 Dynamic 3D Avatar Creation from Hand-held Video Input **7**

2.1 Introduction 8

2.2 Related Work 10

2.3 Overview 13

2.4 Static Modeling 15

2.4.1 Geometry Registration 16

2.4.2 Texture Reconstruction 18

2.5 Dynamic Modeling 19

2.5.1 Reconstructing the Blendshape Model 19

2.5.2 Reconstructing Detail Maps 25

2.6 Animation 28

2.7 Evaluation 29

2.8 Conclusion 34

2.9 Implementation Details 35

3 Semantic Parametric Body Shape Estimation from Noisy RGB-D Sequences **39**

3.1 Introduction and Related Work 40

3.2 Proposed methodology 43

3.2.1 Data Representation 43
3.2.2 Feature Constraints 44
3.2.3 Point-to-plane Constraints 45
3.2.4 Contour Constraints 45
3.2.5 Prior Energy 46
3.2.6 Smoothness Energy 48
3.2.7 Tracking and Modeling 49
3.3 Implementation 52
3.4 Experimental results 52
3.4.1 Effect of each tracking energy 57
3.4.2 Performance Evaluation 59
3.5 Conluding Remarks 59

4 Reconstructing Personalized Anatomical Models for Physics-based Body Animation 61
4.1 Introduction 62
4.2 Related Work 65
4.3 Template Body Model 67
4.4 Forward Skinning Model 69
4.5 Inverse Body Modeling 75
4.5.1 Handling Collisions 77
4.5.2 Registration 78
4.6 Animation 78
4.7 Results 80
4.8 Implementation Details 82
4.9 Limitations and Future Work 85

5 Building and Animating User-Specific Volumetric Face Rigs 87
5.1 Introduction 89
5.2 Related Work 91
5.3 Method 93
5.3.1 Volumetric modeling of actor's neutral face 94
5.3.2 Registration of actor's facial expressions 96
5.3.3 Volumetric facial rigging 100
5.3.4 Animation 100
5.3.5 Collisions 102
5.4 Corrective blendshapes 104
5.5 Implementation and results 105

5.6 Conclusion 108
5.7 Acknowledgements 109

6 Phace: Physics-based Face Modeling and Animation 111
6.1 Introduction 112
6.2 Related Work 116
6.3 Template Face Model 119
6.4 Forward Physics 120
6.5 Inverse Physics 124
6.6 Phace Modeling and Animation 128
6.7 Evaluation 130
6.8 Application Demos 132
6.9 Limitations and Future Work 136
6.10 Conclusion 137

7 Conclusions 139
7.1 Summary 139
7.2 Future Work 140
7.3 Final Remarks 141

Bibliography 158

Curriculum Vitae 159

1 Introduction

1.1 Motivation

The most natural way for humans to communicate is through close face-to-face interactions. As technology evolved, humans grew the need to send information to one another at great distances, without physically traveling themselves. We mention a few approaches in chronological order: written mail, telegraphs, telephones, e-mail, instant messaging, and more recently, video conferencing. All these ways of tackling long-distance human communication do increasingly well at transmitting messages in an efficient manner, but to various extents they lack the emotional presence of face-to-face discussions. In other words, they lack immersion.

While a lot of impressive advancements have been done recently in the fields of virtual, augmented and mixed reality, both in the hardware and software departments, the problem of creating compelling interactive digital human avatars is still elusive. For such new environments, credible 3D avatar representations are of utmost importance.

Throughout this thesis, we identify and propose solutions to two important subproblems in the quest to build and animate realistic digital humans: *lightweight acquisition* and *physics-based animation*.

The first part of the thesis focuses on lightweight acquisition - how to extract as much information as possible from commercially-accessible devices with optical sensors that produce noisy measurements. During this work, we have explored monocular stills and video data coming from a standard mobile phone, as well as low quality RGB-D data recorded using the first mass-market depth sensor, i.e., the Primesense family which includes the Microsoft Kinect.

The second part revolves around employing physics-based models for the reconstruction and animation stages. By extensively using industry-standard linear models for facial and body animation in the previous projects, it has become apparent that these models do not provide the flexibility necessary to obtain highly realistic animations. This becomes particularly interesting when the amount of input data to be used for avatar reconstruction is limited (e.g., a few 3D scans obtained after a short capture session, as opposed to hours of data collection), and user assistance is kept at a minimum.

Data-driven techniques are very effective at using the input information, but most, if not all, discard real-world priors. For example, in the last centuries vast amounts of effort have been put into better understanding the inner workings of the human body. Why should we discard all this well-understood knowledge and rely almost blindly on abstract sensor information we have gathered? The projects in the second half of the thesis show how we can take medical knowledge, build abstract animation models using mechanics principles, and then use those to create compelling novel animations.

It has been shown in the literature that realistic face and body animations can be obtained through various interpolatory techniques, usually based on machine learning concepts. The downside of such methods is that a lot of real-world measurements need to be collected. As expected, this requires long and expensive scanning sessions. We approach the problem from a different perspective. We reduce the amount of data required to be collected by proposing novel physics-based animation models that act as strong regularizers in the reconstruction and animation processes. While they are less lightweight than their data-driven counterparts, they provide excellent extrapolatory properties, especially under secondary motion and previously unseen interaction with external objects and forces.

1.2 Publications

The published work on which this thesis is based on are the following, in chronological order of publication:

- ICHIM, A.E., BOUAZIZ, S., AND PAULY, M. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2015*
- ICHIM, A.E. AND TOMBARI, F. Semantic Parametric Body Shape Estimation from Noisy RGB-D Sequences. *Robotics and Autonomous Systems, 2015*
- ICHIM, A.E., KAVAN, L., NIMIER-DAVID, M., AND PAULY, M. Building and Animating

User-Specific Volumetric Face Rigs. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA), 2016*

- KADLECEK, P.(*), ICHIM, A.E.(*), LIU, T., KAVAN, L., AND KRIVANEK, J. (* joint first authors). Reconstructing Personalized Anatomical Models for Physics-based Body Animation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia), 2016*
- ICHIM, A.E., KADLECEK, P., KAVAN, L., AND PAULY, M. Phace: Physics-based Face Modeling and Animation, *ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2017*

Each chapter starts with an explicit paragraph describing the contributions of the author.

During the PhD work of the author, other papers have been published. However, they have not been included in this thesis:

- HOLZ, D., ICHIM, A.E., TOMBARI, F., RUSU, R.B., AND BEHNKE, S. A modular framework for aligning 3D point clouds - Registration with the Point Cloud Library. *IEEE Robotics and Automation Magazine, 2015*
- ACHILLES, F., ICHIM, A.E., COSKUN, H., TOMBARI, F., NOACHTAR, S., AND NAVAB, N. Patient MoCap: Human Pose Estimation under Blanket Occlusion for Hospital Monitoring Applications. *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), 2016*
- BERTHOLET, P., ICHIM, A.E., AND ZWICKER, M. Temporally Consistent Motion Segmentation from RGB-D Video. *arXiv, August 2016*
- ICHIM, A.E., POPOVIC, J., KAUFMAN, D., WAMPLER, D., AND PAULY, M. Multi-Layer 2D Simulation. *to be published*

1.3 Organization

The thesis presents multiple human face and body reconstruction and performance capture approaches.

- Chapter 2 describes a complete pipeline that allows the user to reconstruct fully rigged, personalized 3D facial avatars using image and video data coming from a hand-held device (e.g., off-the-shelf smartphone). The resulting character mimics the facial expression dynamics of the user by adapting a blendshape template to the recorded video through an optimization that integrates feature tracking, optical flow, and shape from shading. Fine-scale details such as wrinkles are also captured and animated through a learnt regressor. We believe that this system is one of the first to demonstrate that the use of appropriate reconstruction priors yields compelling face rigs even with a minimalistic acquisition system and limited user assistance.

- Continuing along the lines of accessible acquisition systems, Chapter 3 proposes a complete framework for tracking and modeling articulated human bodies from sequences of range maps acquired from inexpensive RGB-D sensors. Our system fits a pre-defined parameteric shape model to depth data by exploiting the simultaneous tracking of the 3D body pose. In addition, compact semantic tags associated to the estimated body shape can be produced by leveraging on an open-source body modeling software and L1 regularization.

- The first two chapters (2 and 3) performed face and body avatar creation by exploiting linear reconstruction and animation models. Such models are not versatile when it comes to incorporating features such as dynamics, collision response, or incompresibility of the flesh. Physics-based models can deliver these effects, but previous approaches lost the controllability that artists were used to with linear models. Chapter 5 proposes a method that combines the benefits of blendshapes with the advantages of physics-based simulation, through the use of novel volumetric blendshapes that are driven by the same weights as traditional blendshapes. We fit our volumetric template model to a set of 3D scans of the actor's face through the usage of physics-inspired 3D fitting priors, and then are able to produce new animation sequences complete with dynamics, secondary motion, collision response and volume preservation through a fast physics simulation.

- Chapter 4 presents a method to create personalized anatomical models ready for physics-based animation using only a set of 3D surface scans. This technique departs from the classical data-driven approaches by using a template anatomical model and physics-based growth constraints. The key contribution is formulating and solving a large-scale optimization problem where subject-specific and pose-dependent parameters are computed such that the resulting anatomical model explains the captured 3D scans as closely as possible. The resulting body avatars are volumetric physics-based models, which provide realistic 3D geometry of the bones and muscles, and support effects such as inertia, gravity, and collisions according to Newtonian dynamics.

- Chapter 6 extends the inverse physics formulation presented in Chapter 4 for usage with facial animation. While bodies were actuated by a softly-coupled rigid skeleton, faces are actuated by both bones (cranium and mandible), as well as muscles. As such, we compute facial expressions by minimizing a set of non-linear potential energies that model the physical interaction of passive flesh, active muscles, and rigid bone structures. A novel muscle activation model leads to a robust optimization that faithfully reproduces complex facial articulations. Our method supports temporal dynamics due to inertia or external forces, incorporates skin sliding to avoid unnatural stretching, and offers full control of the simulation parameters, which enables a variety of advanced animation effects. For example, slimming and fattening is achieved by scaling the volume of the soft tissue elements. This approach is explified with multiple demos, including artistic editing of the animation model, simulation of corrective facial surgery, or dynamic interaction with external forces and objects.

At the end of each chapter there is a section called *Retrospective* that brings an updated view on the projects presented in each chapter, linking them together, as well as including comments referring to newer published related work.

2 Dynamic 3D Avatar Creation from Hand-held Video Input

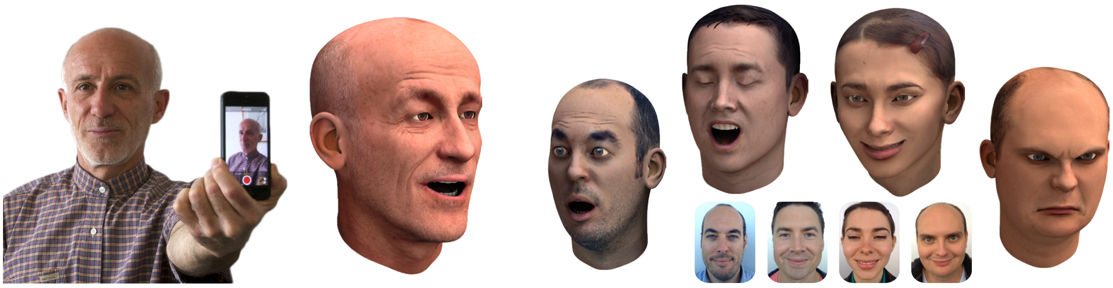


Figure 2.1 – Our system creates a fully rigged 3D avatar of the user from uncalibrated video input acquired with a cell-phone camera. The blendshape models of the reconstructed avatars are augmented with textures and dynamic detail maps, and can be animated in realtime.

Note

This chapter is based on the following publication [IBP15]:

ICHIM, A.E., BOUAZIZ, S., AND PAULY, M. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2015*

The candidate contributed with most of the scientific contributions and implementation of this publication.

Abstract

We present a complete pipeline for creating fully rigged, personalized 3D facial avatars from hand-held video. Our system faithfully recovers facial expression dynamics of the user by

adapting a blendshape template to an image sequence of recorded expressions using an optimization that integrates feature tracking, optical flow, and shape from shading. Fine-scale details such as wrinkles are captured separately in normal maps and ambient occlusion maps. From this user- and expression-specific data, we learn a regressor for on-the-fly detail synthesis during animation to enhance the perceptual realism of the avatars. Our system demonstrates that the use of appropriate reconstruction priors yields compelling face rigs even with a minimalistic acquisition system and limited user assistance. This facilitates a range of new applications in computer animation and consumer-level online communication based on personalized avatars. We present realtime application demos to validate our method.

2.1 Introduction

Recent advances in realtime face tracking enable fascinating new applications in performance-based facial animation for entertainment and human communication. Current realtime systems typically use the extracted tracking parameters to animate a set of pre-defined characters [WLVGP09, WBLP11, LYYB13, CWLZ13, BWP13, CHZ14]. While this allows the user to enact virtual avatars in realtime, personalized interaction requires a custom rig that matches the facial geometry, texture, and expression dynamics of the user. With accurate tracking solutions in place, creating compelling user-specific face rigs is currently a major challenge for new interactive applications in online communication. In this paper we propose a software pipeline for building fully rigged 3D avatars from hand-held video recordings of the user.

Avatar-based interactions offer a number of distinct advantages for online communication compared to video streaming. An important benefit for mobile applications is the significantly lower demand on bandwidth. Once the avatar has been transferred to the target device, only animation parameters need to be transmitted during live interaction. Bandwidth can thus be reduced by several orders of magnitude compared to video streaming, which is particularly relevant for multi-person interactions such as conference calls.
A second main advantage is the increased content flexibility. A 3D avatar can be more easily integrated into different scenes, such as games or virtual meeting rooms, with changing geometry, illumination, or viewpoint. This facilitates a range of new applications, in particular on mobile platforms and for VR devices such as the Occulus Rift.

Our goal is to enable users to create fully rigged and textured 3D avatars of themselves at home. These avatars should be as realistic as possible, yet lightweight, so that they can be readily integrated into realtime applications for online communication. Achieving this goal implies meeting a number of constraints: the acquisition hardware and process need to be

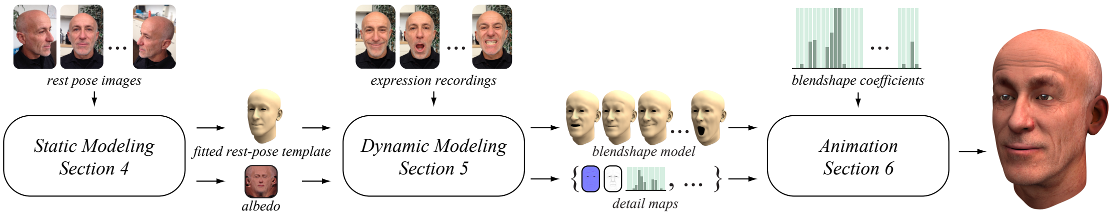


Figure 2.2 – The main stages of our processing pipeline. *Static Modeling* reconstructs the geometry and albedo of the neutral pose. *Dynamic Modeling* adapts a generic blendshape model to the recorded user and reconstructs detail maps for each video frame. *Animation* drives the reconstructed rig using blendshape coefficients and synthesizes new pose-specific detail maps on the fly.

simple and robust, precluding any custom-build setups that are not easily deployable. Manual assistance needs to be *minimal* and restricted to operations that can be easily performed by untrained users. The created rigs need to be *efficient* to support realtime animation, yet accurate and detailed to enable engaging virtual interactions.

These requirements pose significant technical challenges. We maximize the potential user base of our system by only relying on simple photo and video recording using a hand-held cell-phone camera to acquire user-specific data.
The core processing algorithms of our reconstruction pipeline run automatically. To improve reconstruction quality, we integrate a simple UI to enable the user to communicate tracking errors with simple point clicking. User assistance is minimal, however, and required less than 15 minutes of interaction for all our examples.
Realtime applications are enabled by representing the facial rig as a set of blendshape meshes with low polygon count. Blendshapes allow for efficient animation and are compatible with all major animation tools. We increase perceptual realism by adding fine-scale facial features such as dynamic wrinkles that are synthesized on the fly during animation based on precomputed normal and ambient occlusion maps.

We aim for the best possible quality of the facial rigs in terms of geometry, texture, and expression dynamics. To achieve this goal, we formulate dynamic avatar creation as a geometry and texture reconstruction problem that is regularized through the use of carefully designed facial priors. These priors enforce consistency and guarantee a complete output for a fundamentally ill-posed reconstruction problem.

Contributions. We present a comprehensive pipeline for video-based reconstruction of fully-rigged, user-specific 3D avatars for consumer applications in uncontrolled environments. Our core technical contributions are:

- a two-scale representation of a dynamic 3D face rig that enables realtime facial animation by integrating a medium-resolution blendshape model with a high-resolution albedo map and dynamic detail maps;
- a novel optimization method for reconstructing a consistent albedo texture from a set of input images that factors out the incident illumination;
- a new algorithm to build the dynamic blendshape rig from video input using a joint optimization that combines feature-based registration, optical flow, and shape-from-shading;
- an offline reconstruction and online synthesis method for fine-scale detail stored in pose-specific normal and ambient occlusion maps.

We demonstrate the application potential of our approach by driving our reconstructed rigs both in a realtime animation demo and using a commodity performance capture system. With our minimalistic acquisition setup using only a single cellphone camera, our system has the potential to be used by millions of users worldwide.

2.2 Related Work

We provide an overview of relevant techniques for 3D facial avatar creation. We start by covering techniques for high quality *static* modeling of human faces. We then discuss approaches that attempt to capture fine-scale information associated with *dynamic* facial deformations, like expression lines and wrinkles. Finally, as our target is the creation of an animatable avatar, we will also discuss methods that attempt to map the acquired dynamic details onto given input animation data.

Static modeling. Due to the high complexity of facial morphology and heterogeneous skin materials, the most common approaches in facial modeling are data-driven. The seminal work of [BV99] builds a statistical (PCA) model of facial geometry by registering a template model to a collection of laser scans. Such a model can be employed to create static avatars from a single image [BV99] or from multiple images [ABF*07, DIF04], or for the creation of personalized real-time tracking profiles [WBLP11, LYYB13, BWP13]. However, as a compact PCA model only captures the coarse-scale characteristics of the dataset, the generated avatars are typically rather smooth, lacking the ability to represent fine-scale features like wrinkles and expression lines.

Fine-scale detail for facial modeling has been recovered in a controlled environment with multiple calibrated DSLR cameras in the work of Beeler et al. [BBB*10]. This setup allows capturing wrinkles, skin pores, facial hair [BBN*12], and eyes [BBN*14]. The more involved system of [GFT*11] uses fixed linear polarizers in front of the cameras and enables accurate acquisition of diffuse, specular, and normal maps. While effective for high-end productions, such systems require a complex calibration within a lab environment and are thus unsuitable for personalized avatar creation at home. In contrast, our approach uses only a cell-phone camera, requires neither calibration nor a controlled environment, and only relies on minimal user assistance.

Dynamic modeling. A static reconstruction only recovers the geometry and texture for a single facial expression. To build compelling avatars, we also need to reconstruct a dynamic expression model that faithfully captures the user's specific facial movements. One approach to create such a model is to simulate facial muscle activation and model the resulting bone movements and viscoelastic skin deformations [VLR05, WKT96]. However, the large computational cost and complex parameter estimation make such an approach less suitable for facial animation.

Consequently, parametric models are typically employed to represent dynamic skin behavior [Oat07, JEOG11]. Unfortunately, such models are not only difficult to design, but are typically also custom-tuned to a particular animation rig. This makes it difficult to infer generic models for facial dynamics that can easily be adapted to specific subjects. For these reasons, data-driven techniques are again the most common way to approach the reconstruction of facial dynamics.

The multi-linear models introduced by [VBPP05] and then further explored in [CWZ*14] offer a way of capturing a joint space of pose and identity. Alternatively, rather than assuming an offline prior on pose and identity, dynamic geometry variations can be linearly modeled in realtime while tracking videos [CHZ14] or RGB-D data [BWP13, LYYB13]. These compact linear models are tailored towards estimating a small set of tracking parameters to enable realtime performance, and consequently are not suitable to recover detailed avatars. Our approach builds upon this prior work, but utilizes detail information in the acquired images to recover a significantly richer set of facial features for avatar creation.

The use of custom hardware has been the most successful way of estimating dynamic

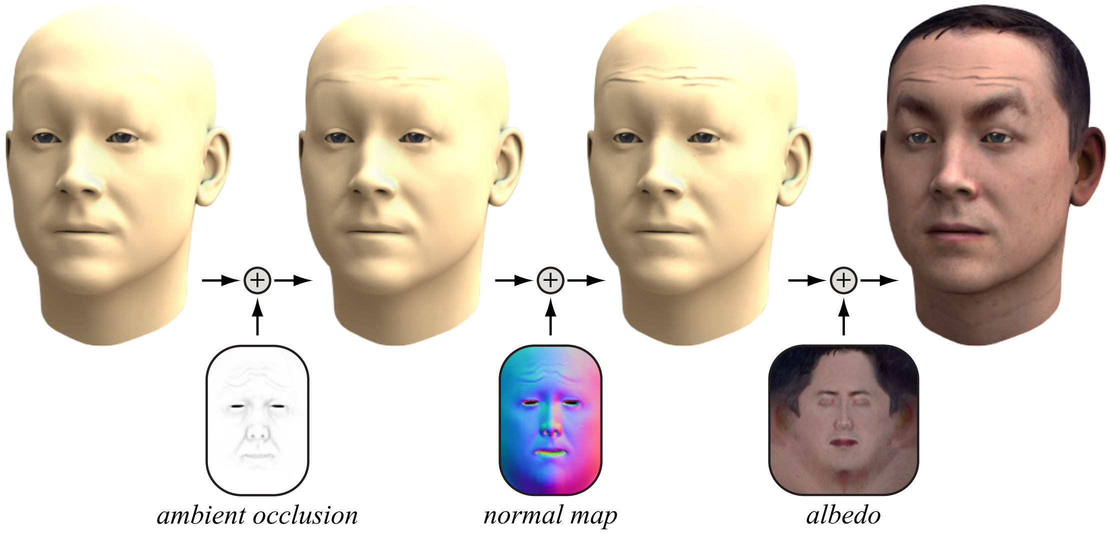


Figure 2.3 – Our dynamic face rig augments a low-resolution blendshape pose (left) with dynamic per-texel ambient occlusion coefficients and normals, and a high-resolution albedo texture.

avatars for high-end productions. For example, the Digital Emily project [ARL*09] demonstrates how the Light Stage system enables photorealistic dynamic avatars. The work of Alexander et al. [AFB*13] recently extended this approach to enable real-time rendering of highly detailed facial rigs. Structured light and laser scanners have also been used to acquire facial geometry at the wrinkle scale [ZSCS04, MJC*08, LAGP09, HCTW11]. Similarly, the setup of [BBB*10, BHB*11] is capable of reconstructing fine-scale detail using multiple calibrated/synchronized DSLR cameras. More recent work attempts to further reduce the setup complexity by only considering a *binocular* [VWB*12] or a hybrid *binocular/monocular* setup [GVWT13]. We push this trend to its limit by only requiring hand-held video recording in an uncontrolled environment.

Animation. While the methods above are able to infer detailed geometry we aim for the creation of an avatar of the recorded user, that can be animated programmatically or using other sources of tracking parameters. The systems of [GVWT13], and [SWTC14] essentially recover detailed facial geometry by providing one mesh per frame deformed to match the input data. The former uses a pre-built user-specific blendshape model for the face alignment by employing automatically corrected feature points [SLC11]. A dense optical flow field is computed in order to smoothly deform the tracked mesh at each frame, after which a shape-from-shading stage adds high frequency details. Although our tracking approach and detail enhancement is based on similar principles, the aim of our approach is to integrate all these shape corrections directly into our proposed two-scale representation of dynamic 3D faces. Shi et al. [SWTC14] use their own feature detector along with a non-rigid structure-from-motion algorithm to track and model the identity and per-frame expressions of the face by employing a bilinear face model. Additionally, a keyframe-based iterative approach using shape from shading is employed in order to further refine the bilinear model parameters, as well as the albedo texture of the face, and per-frame normal maps exhibiting high frequency details such as wrinkles Neither method aims at creating an animation-ready avatar that incorporates all of the extracted details.

Of the methods presented above, only Alexander and colleagues [ARL*09, AFB*13] produce a blendshape model that can be directly embedded in animation software, but as mentioned, the complexity of the setup makes it unsuitable for consumer applications. The recent techniques of [BBB*14], and [LXC*15] can re-introduce high frequency details in a coarse input animation, if a high-resolution performance database is provided. Conversely, our technique generates an animatable blendshape model augmented with dynamic detail maps using only consumer camera data. Our rigged avatars can thus be directly driven by tracking software, e.g. [WBLP11, SLC11], or posed in a keyframe animation system.

2.3 Overview

We first introduce our two-scale representation for 3D facial expression rigs. Then we discuss the acquisition setup and provide a high-level description of the main processing steps for the reconstruction and animation of our rigs (Figure 2.2). The subsequent sections explain the core technical contributions, present results, and provide an evaluation of our method. The paper concludes with a discussion of limitations and an outline of potential future work. Implementation details are provided in the Appendix.

Dynamic Face Rig. Our method primarily aims at the reconstruction of 3D facial rigs for realtime applications. We therefore propose a two-scale representation that strikes a balance between a faithful representation of the dynamic expression space of the recorded user and the efficient animation of the reconstructed avatar. This balance can be achieved using a coarse blendshape mesh model of approximatively 10k vertices that is personalized to the specific user and augmented with texture and detail information as shown in Figure 2.3.

A specific facial expression is represented by a linear combination of a set of blendshapes [LAR*14]. At low resolution, the blendshape representation can be efficiently evaluated

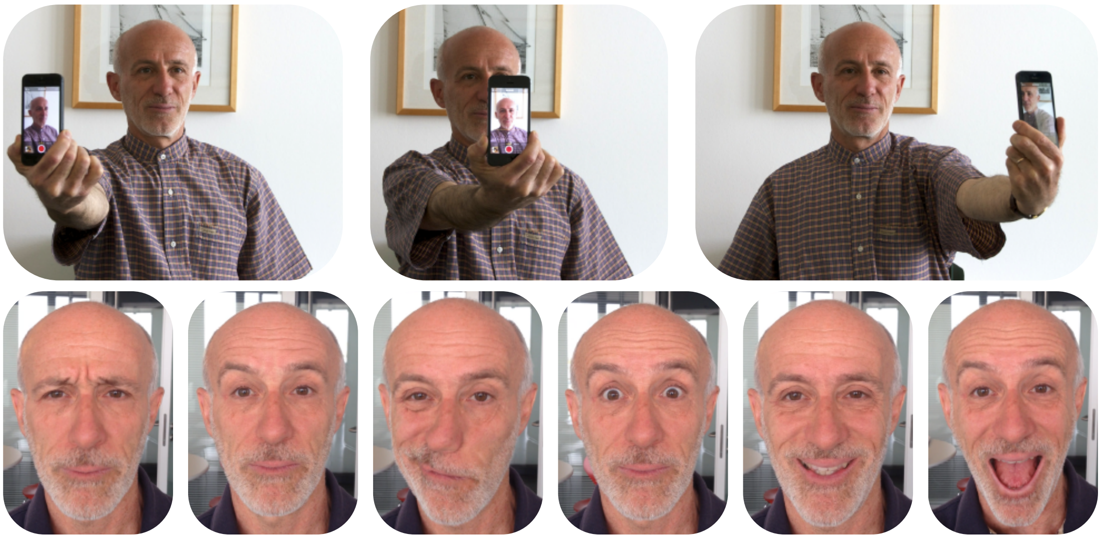


Figure 2.4 – All acquisition is performed with a hand-held cell phone camera. A semi-circular sweep is performed for the static reconstruction (top row), a frontal video is recorded for the dynamic modeling (bottom row).

and rendered, but lacks fine-scale detail. We therefore augment the mesh with a static high-resolution albedo map to capture color variations across the face. In addition, we build dynamic high-resolution maps with per-pixel normals and ambient occlusion coefficients to represent fine-scale geometric features. We refer to these latter maps as *detail maps* in the subsequent text. Previous methods such as [BLB*08] use a similar two-scale decomposition, but operate on high-resolution meshes and can thus represent details as displacements. To avoid the complexities of realtime displacement mapping we opted for normal and ambient occlusion maps that can be synthesized and rendered more efficiently during animation.

Acquisition. In order to build a dynamic rig of the user we need to capture enough information to reconstruct the blendshapes, the albedo texture, and the detail maps. At the same time, keeping our consumer application scenario in mind, we want to restrict the acquisition to simple hardware and a minimalistic process that can be robustly performed by anyone. We therefore opted for a simple hand-held cell-phone camera. The user first records her- or himself in neutral expression by sweeping the camera around the face capturing images in burst mode. We then ask the user to record a video in a frontal view while performing different expressions to capture user-specific dynamic face features (see Figure 2.4). For all our acquisitions we use an Apple iPhone 5 at 8 megapixel resolution for static photo capture and 1080p for dynamic video recordings (see accompanying video).

The key advantage of our acquisition setup is that we do not require any calibration, synchronization, or controlled lighting. All acquisitions can be done by an inexperienced user in approximatively 10 minutes. However, this simplistic acquisition process poses significant challenges for our reconstruction algorithms as the quality of the input data is significantly lower than for existing calibrated studio setups.

Processing Pipeline. Figure 2.2 provides an overview of our processing pipeline. We split the reconstruction into a static and a dynamic modeling stage. In the static stage (Section 2.4) we first reconstruct a 3D point cloud from the photos taken in neutral pose using a multi-view stereo algorithm. We then apply non-rigid registration to align a template mesh to this point cloud to model the user's face geometry. A static albedo texture is extracted by integrating the color images into a consistent texture map.

The dynamic modeling stage (Section 2.5) reconstructs expression-specific information. Given the neutral pose, we first transfer the deformations of a generic blendshape model to obtain an initial blendshape representation for the user. We further refine this user-specific blendshape model using an optimization that integrates texture-based tracking and shading cues to best match the geometric features of the recorded user. The reconstructed blendshape model then faithfully recovers the low- and medium frequency dynamic geometry of the user's face. However, high frequency details such as wrinkles are still missing from the rig. In order to capture these details we automatically extract a set of dynamic detail maps from the recorded video frames.

Finally, in the animation stage (Section 2.6), the reconstructed rig can be driven by a temporal sequence of blendshape coefficients. These animation parameters can either be provided manually through interactive controllers, or transferred from a face tracking software. The specific detail map for each animated pose of the avatar is synthesized on the fly from the captured detail maps using a trained regressor driven by surface strain.

2.4 Static Modeling

This section describes the static modeling stage of the reconstruction pipeline (see Figure 2.5). The first part of the acquisition provides us with a set of images of the user in neutral expression from different viewpoints. From these uncalibrated images we extract a point cloud using a state-of-the-art structure from motion (SFM) software [FP10, Wu13]. We then use a geometric morphable model [BV99], representing the variations of different human faces in neutral

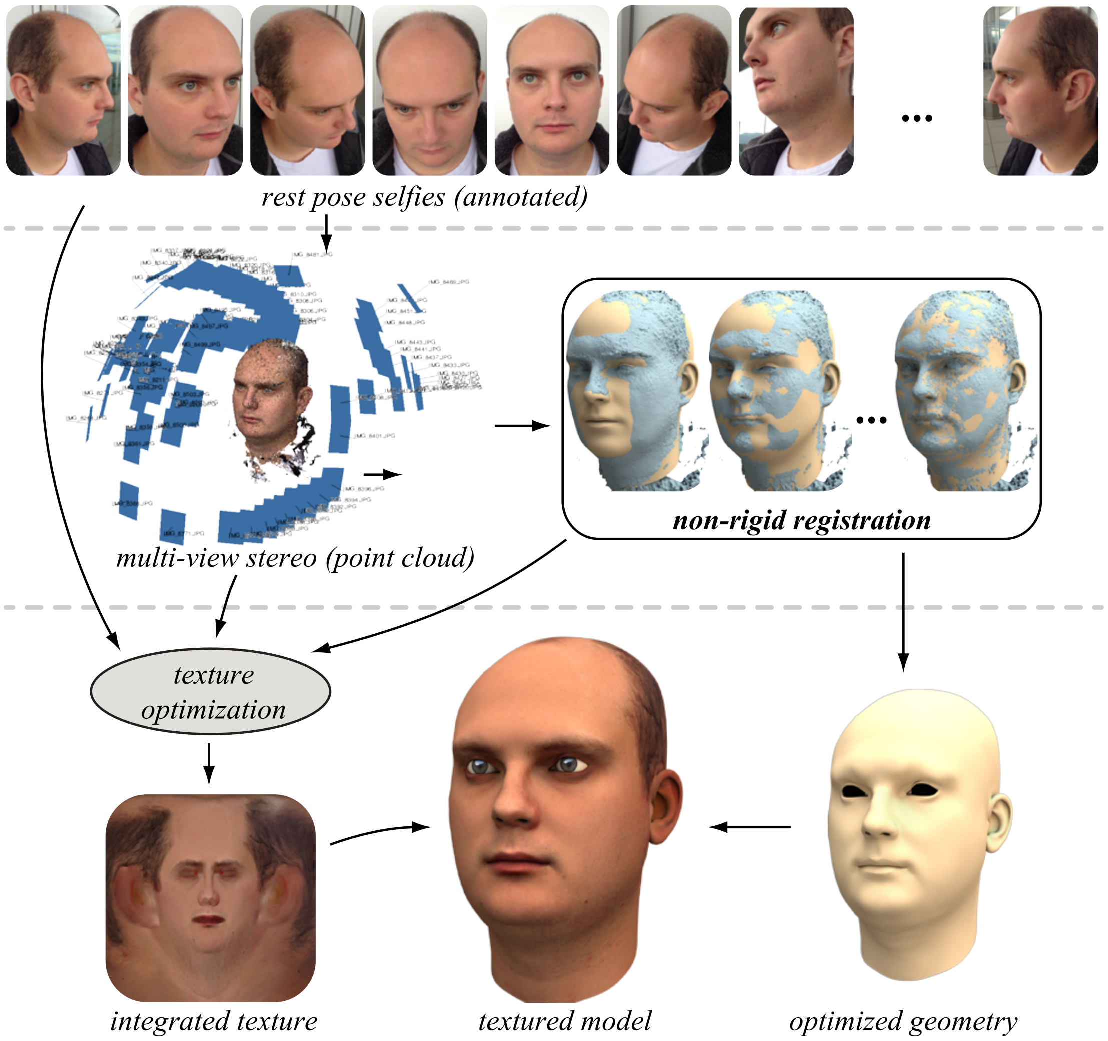


Figure 2.5 – Static Modeling recovers the neutral pose. A deformable template model is registered to a 3D point cloud computed from a set of images using multi-view stereo. A static albedo texture integrates color information of all recorded images while factoring out the illumination.

expression, as a prior for reconstruction.

2.4.1 Geometry Registration

We register the morphable model towards the point cloud to obtain a template mesh that roughly matches the geometry of the user's face. We improve the registration accuracy using non-rigid registration based on thin-shell deformation [BKP*10, BTP14].

The registration is initialized by using 2D-3D correspondences of automatically detected

2D facial features [SLC11] in each input frame. For the precise non-rigid alignment of the mouth, eye and eyebrow regions, the user is asked to mark a few contours in one of the frontal images as illustrated on the right.

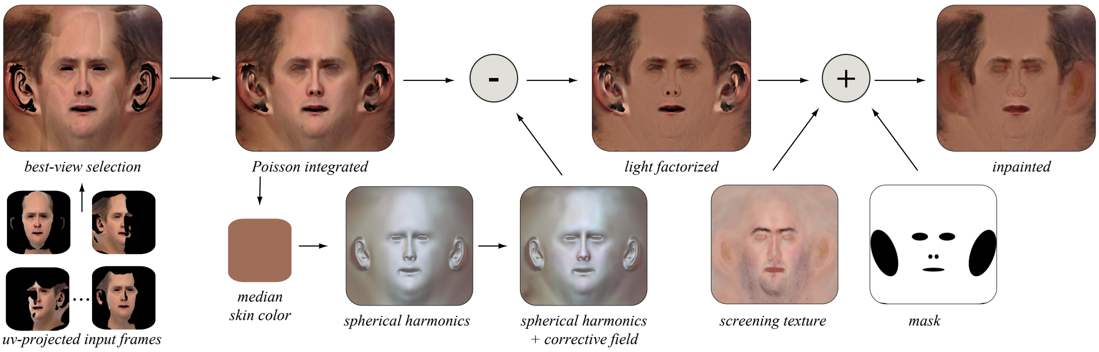
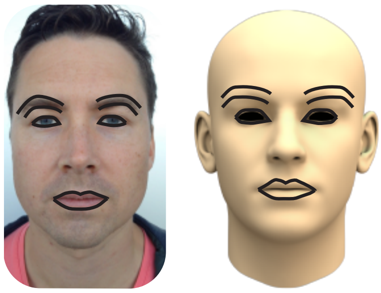


Figure 2.6 – Reconstructing the albedo map. Poisson integration combines the pixel colors of different input images using a gradient optimization. The illumination is factored out based on a lighting model that combines a 2nd order spherical harmonics approximation with a per-pixel corrective field. The screening texture provides a reconstruction prior to complete missing parts in the albedo map.



To improve the realism of the reconstructed avatars, we add eyes and inner mouth components, i.e., teeth, tongue, and gums. These parts are transferred from the template model and deformed to match the reconstructed head geometry by optimizing for the rotation, translation and anisotropic scaling using a set of predefined feature points around the mouth and eye regions. We also adapt the texture for the eye meshes to the recorded user. The iris is found by detecting the largest ellipse inside the projection of the eye region to the most frontal input image using Hough transform [DH72]. Histogram matching is performed between a template eye texture and the image patch corresponding to the iris [GW06]. The images in Figure 2.7 illustrate the eye texture adaptation for one example subject.

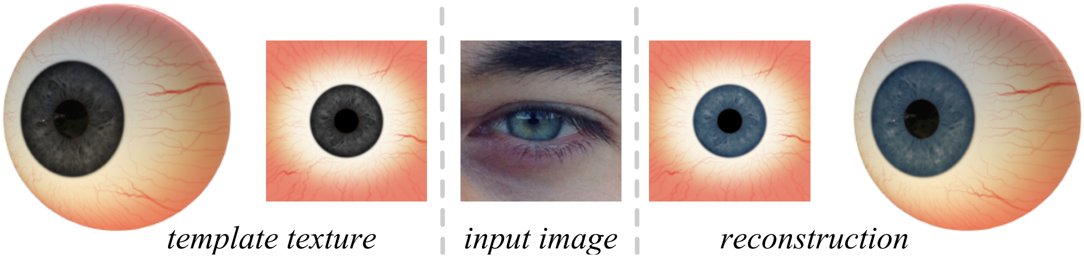


Figure 2.7 – Our approach for eye texture adaptation for one example subject.

We currently do not recover the specific geometry or appearance of the user's teeth or tongue, which is an interesting topic for future work.

2.4.2 Texture Reconstruction

Given the registered template mesh, the next step in the pipeline is the reconstruction of a high-resolution albedo texture (see Figure 2.6). We use the UV parameterization of the template to seamlessly combine all images using Poisson integration [PGB03]. This is achieved by selecting the color gradients of the pixels with the most parallel view rays to the surface normals.

Factorizing Illumination. After integration, the texture map not only contains the RGB reflectance but also the specific illumination of the recording environment. This may be problematic as the final mesh will be used in a virtual environment where the lighting may not match the one baked into the texture. To factor out the illumination from the skin albedo, we define the color of a skin pixel $\{i, j\}$ as $\mathbf{c}_{ij} = \mathbf{r}_{ij} \circ \mathbf{s}_{ij}$, where $\mathbf{r}_{ij}$ is the skin reflectance, $\mathbf{s}_{ij}$ accounts for the illumination, and $\circ$ denotes the entry-wise product. We assume a smooth illumination and we represent it using spherical harmonics. Low-dimensional lighting representations using spherical harmonics are effective in numerous lighting situations with a variety of object geometries [FSB04]. However, they are not expressive enough to account for complex conditions involving self-shadowing or complex specularities. This is due to the fact that spherical harmonics have the limitation of being only expressed as a function of the surface normals, i.e., points with similar normals will have a similar illumination. To compensate for the inaccuracy of this illumination model, we augment the spherical harmonics with corrective fields in uv-space $\mathbf{d}_{ij} = [d^r_{ij} \;\; d^g_{ij} \;\; d^b_{ij}]$ for the R, G and B color channel, respectively. This leads to

$$\mathbf{s}_{ij} = \mathbf{y}^T \phi(\mathbf{n}_{ij}) + \mathbf{d}^T_{ij}, \tag{2.1}$$

where $\mathbf{n}_{ij}$ is the mesh normal at the pixel p and

$$\phi(\mathbf{n}) = [1, n_x, n_y, n_z, n_x n_y, n_x n_z, n_y n_z, n_x^2 - n_y^2, 3n_z^2 - 1]^T \tag{2.2}$$

is a vector of second order spherical harmonics with corresponding weight vectors $\mathbf{y} = [\mathbf{y}^r \;\; \mathbf{y}^g \;\; \mathbf{y}^b]$. As the illumination is assumed to be of low frequency, we require the corrective fields to be smooth. In addition, we assume that the required corrections are small. This leads to a minimization over the spherical harmonics weight vectors and the corrective fields expressed as

$$\min_{\mathbf{y},\mathbf{d}} \sum_{i,j} \|\mathbf{r} \circ \mathbf{s}_{ij} - \mathbf{c}_{ij}\|_2^2 + \lambda_1 \|\mathbf{d}\|_F^2 + \lambda_2 \|\mathbf{G}\mathbf{d}\|_F^2 + \lambda_3 \|\mathbf{L}\mathbf{d}\|_F^2, \tag{2.3}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathbf{d}$ stacks all the $\mathbf{d}_{ij}$, $\mathbf{G}$ is the gradient matrix, and $\mathbf{L}$ is the graph Laplacian matrix. Both the gradient and Laplacian are computed with periodic boundary condition. The non-negative weights λ_1, λ_2, and λ_3 control the magnitude and the smoothness of the corrective fields. To optimize Equation 2.3, we employ a two-stage process, where the skin reflectance is set to a constant $\mathbf{r}$ using the median color of the face pixels. We first compute the spherical harmonics weight vectors by initializing the corrective fields to zero and only optimizing over $\mathbf{y}$. This only requires solving a 9×9 linear system. We then solve for the corrective fields keeping the weight vectors fixed. This minimization can be performed efficiently using a Fast Fourier Transform (FFT) as the system matrix is circulant [Gra06].

We use the extracted illumination $\mathbf{s}_{ij}$ to reconstruct the illumination-free texture. Finally, to generate a complete texture map, we reintegrate the illumination-free texture into the template texture map using Poisson integration. Because the extracted illumination is smooth, i.e., of low frequency, high frequency details are preserved in the final albedo texture (see Figure 2.6).

2.5 Dynamic Modeling

The goal of the dynamic modeling phase is to complete the face rig by reconstructing user-specific blendshapes as well as dynamic normal and ambient occlusion maps (see Figure 2.8). We focus here on the general formulation of our optimization algorithm and refer to the appendix for more details on the implementation.

2.5.1 Reconstructing the Blendshape Model

The blendshape model is represented as a set of meshes $\mathbf{B} = [\mathbf{b}_0, \ldots, \mathbf{b}_n]$, where $\mathbf{b}_0$ is the neutral pose and the $\mathbf{b}_i, i > 0$ are a set of predefined facial expressions. A novel facial expression is generated as $\mathbf{F}(\mathbf{B}, \mathbf{w}) = \mathbf{b}_0 + \Delta\mathbf{B}\mathbf{w}$, where $\Delta\mathbf{B} = [\mathbf{b}_1 - \mathbf{b}_0, \ldots, \mathbf{b}_n - \mathbf{b}_0]$, and $\mathbf{w} = [w_1, \ldots, w_n]^T$ are blendshape weights. The reconstruction prior at this stage is a generic blendshape model consisting of 48 blendshapes (see also accompanying material). We denote with $\mathbf{F}_\mathbf{T}$ the facial expression $\mathbf{F}$ transformed by the rigid motion $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ with rotation $\mathbf{R}$ and translation $\mathbf{t}$.

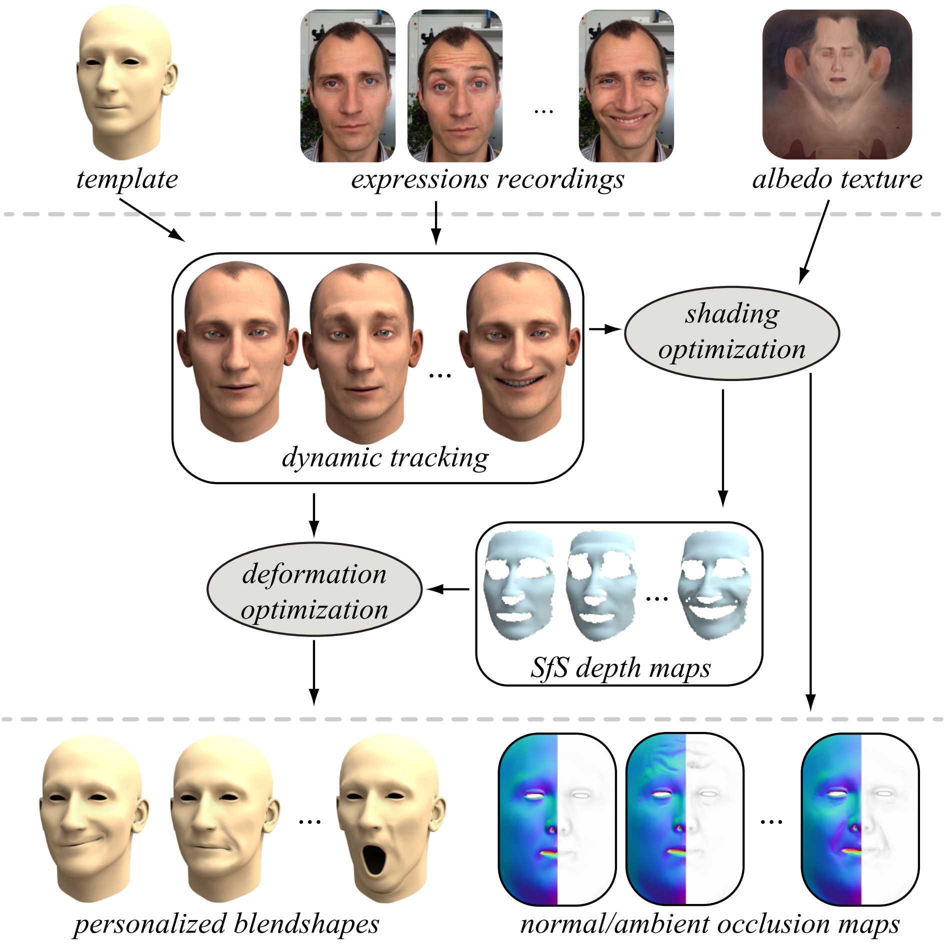


Figure 2.8 – Dynamic Modeling adapts a generic blendshape model to the facial characteristics of the user and recovers expression-specific detail maps from the recorded video sequence.

We initialize the user-specific blendshape model by applying deformation transfer [SP04] from the generic blendshape template to the reconstructed mesh of the user's neutral pose. Deformation transfer directly copies the deformation gradients of the template without accounting for the particular facial expression dynamics of the user. To personalize the blendshape model, we optimize for additional surface deformations of each blendshape to better match the facial expressions of the user in the recorded video sequence. Previous methods, such as [BWP13, LYYB13], perform a similar optimization using 3D depth-camera input. However, these methods only aim at improving realtime tracking performance and do not recover detailed rigged avatars. Moreover, in our reconstruction setting we are not constrained to realtime performance and can thus afford a more sophisticated optimization specifically designed for our more challenging 2D video input data.

Our algorithm alternates between *tracking*, i.e., estimating the blendshape weights and rigid pose of the facial expressions in the image sequence, and *modeling*, i.e., optimizing the blendshapes to better fit the user's expression.

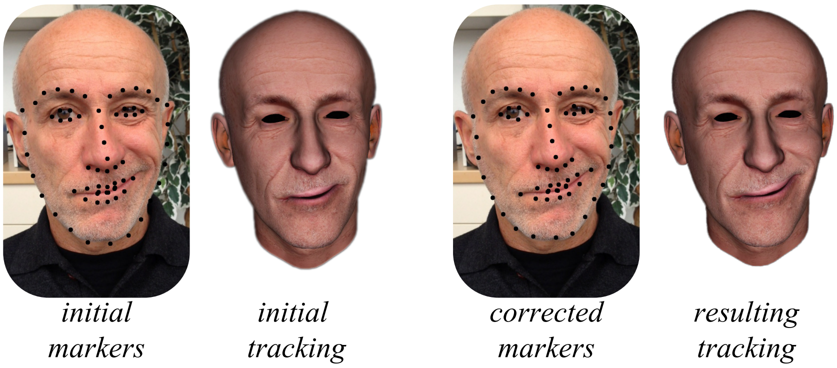

initial markers *initial tracking* *corrected markers* *resulting tracking*

Figure 2.9 – The initial marker locations extracted using feature detection can be corrected by the user to improve the tracking. Here the markers at the mouth corners and jaw line have been manually displaced. Such user annotations are propagated through the entire sequence, so that only a small number of frames need to be corrected.

Tracking. We propose a tracking algorithm using 2D image-based registration based on a combination of feature alignment and optical flow. This results in a per-frame optimization over the blendshape weights $\mathbf{w}$ and the rigid motion $\mathbf{T}$ expressed as

$$\arg\min_{\mathbf{w},\mathbf{T}} E_{\text{feature}} + E_{\text{flow}} + E_{\text{sparse}}. \tag{2.4}$$

We formulate the facial feature energy as

$$E_{\text{feature}} = \gamma_1 \sum_{v \in \mathscr{M}} \|\mathbf{m}_v - P(\mathbf{F}_{\mathbf{T}}(\mathbf{B}_v, \mathbf{w}))\|_2^2, \tag{2.5}$$

where $\mathscr{M}$ is the set of points representing the facial feature locations on the mesh surface, $\mathbf{m}_v$ is the 2D image location of the feature point v extracted using the method of [SLC11], $P(\cdot)$ projects a 3D point to 2D, and $\mathbf{B}_v = \mathbf{c}_v^T \mathbf{B}$, where the vector $\mathbf{c}_v$ contains the barycentric coordinates corresponding to v.

The feature extraction algorithm of [SLC11] is fairly accurate, but does not always find the correct marker locations. To improve the quality of the tracking, we ask the user to correct marker locations in a small set of frames (see Figure 2.9). Following [GVWT13], these edits are then propagated through the image sequence using frame-to-frame optical flow [ZPB07]. For a sequence of 1500 frames, we typically require 25 frames to be manually corrected. With more sophisticated feature extraction algorithms such as [CHZ14], this manual assistance can potentially be dispensed with completely.

To complement the feature energy, we use a texture-to-frame optical flow using a gradient-based approach. This formulation increases the robustness to lighting variations between the static and dynamic acquisition. This energy is defined as

$$E_{\text{flow}} = \gamma_2 \sum_{v \in \mathcal{O}} \left\| \begin{bmatrix} \rho_{v+\Delta v_x} - \rho_v \\ \rho_{v+\Delta v_y} - \rho_v \end{bmatrix} - \begin{bmatrix} I(\mathbf{u}_{v+\Delta v_x}) - I(\mathbf{u}_v) \\ I(\mathbf{u}_{v+\Delta v_y}) - I(\mathbf{u}_v) \end{bmatrix} \right\|_2^2, \tag{2.6}$$

where $\mathcal{O}$ is the set of visible points located on the mesh surface involved in the optical flow constraint, and $\mathbf{u}_v = P(\mathbf{F_T}(\mathbf{B}_v, \mathbf{w}))$. Δv_x is a 3D displacement along the surface such that the surface point $v + \Delta v_x$ maps to the texture pixel immediately above the one corresponding to point v; analogously, $v + \Delta v_y$ maps to the texture pixel on the right. ρ_v is the grayscale value for the point v extracted from the albedo texture, and $I(\mathbf{x})$ is the grayscale color extracted from the image at location $\mathbf{x}$.

We apply an ℓ_1-norm regularization on the blendshape coefficients using

$$E_{\text{sparse}} = \gamma_3 \|\mathbf{w}\|_1. \tag{2.7}$$

This sparsity-inducing term stabilizes the tracking and avoids too many blendshapes being activated with a small weight. Compared to the more common ℓ_2 regularization, this better retains the expression semantics of the blendshape model and thus simplifies tracking and retargeting as shown in [BWP13]. Similar to [WBLP11], we alternate between optimizing for the rigid transformation $\mathbf{T}$ and the blendshape weights $\mathbf{w}$.

Modeling. After solving for the tracking parameters, we keep these fixed and optimize for the vertex positions of the blendshapes. We again use facial features and optical flow leading to

$$\arg\min_{\mathbf{B}} E_{\text{feature}} + E_{\text{flow}} + E_{\text{close}} + E_{\text{smooth}}. \tag{2.8}$$

The closeness term penalizes the magnitude of the deformation from the initial blendshapes $\mathbf{B}^*$ created using deformation transfer:

$$E_{\text{close}} = \gamma_4 \|\mathbf{B} - \mathbf{B}^*\|_F^2. \tag{2.9}$$

The smoothness term regularizes the blendshapes by penalizing the stretching and the bending of the deformation:

$$E_{\text{smooth}} = \gamma_5 \|\mathbf{G}(\mathbf{B} - \mathbf{B}^*)\|_F^2 + \gamma_6 \|\mathbf{L}(\mathbf{B} - \mathbf{B}^*)\|_F^2. \tag{2.10}$$

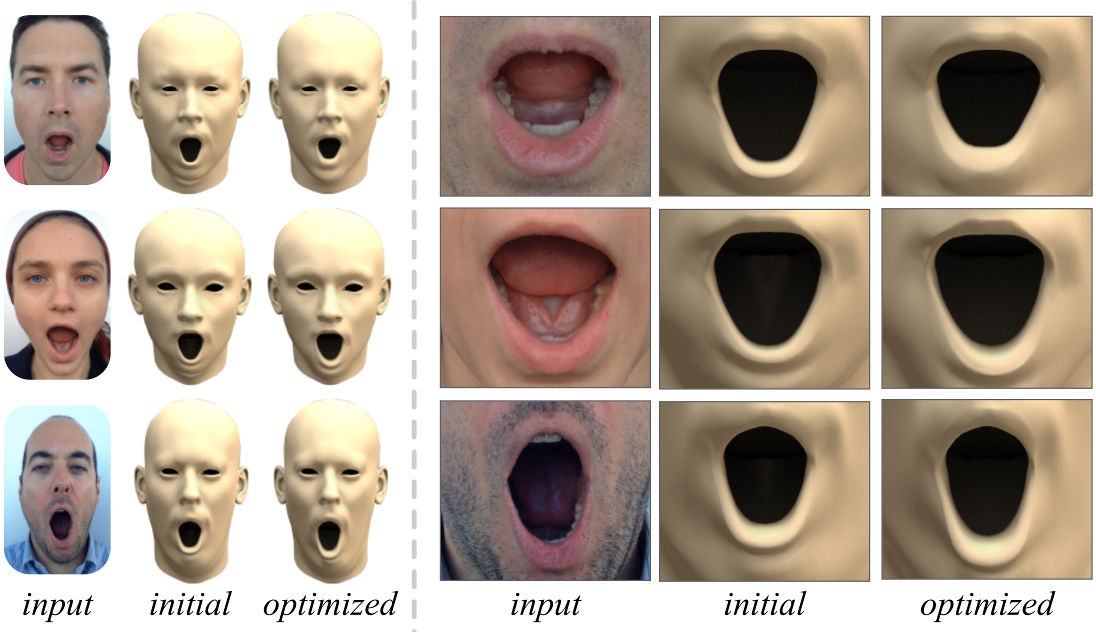


Figure 2.10 – Optimizing blendshapes is essential to accurately represent user-specific expressions. The initial blendshapes computed with deformation transfer (middle column) are registered towards the input images (left column). The resulting optimized blendshapes (right column) faithfully capture expression asymmetries.

In contrast to the tracking optimization of Equation 2.4 that is performed separately for each frame, the blendshape modeling optimization is performed jointly over the whole sequence. Tracking and modeling are iterated 20 times for all our examples.

Geometric Refinement. The blendshape modeling optimization from 2D images is effective for recovering the overall shape of the user-specific facial expressions (see Figure 2.10). We further improve the accuracy of the blendshapes using a 3D refinement step. For this purpose we extract one depth map per frame using a photometric approach [KSB11, WZN*14]. The input video is downsampled by a factor of 8 to a resolution of 150 × 240 pixels in order to capture only the medium-scale details corresponding to the mesh resolution of the blendshape model (see Figure 2.11). Fine-scale detail recovery will be discussed in Section 2.5.2.

For each video frame we rasterize the tracked face mesh recovered during the blendshape optimization to obtain the original 3D location $\bar{\mathbf{p}}_{ij}$ in camera space and the grayscale albedo value ρ_{ij} of each pixel $\{i, j\}$. We compute smooth interpolated positions using cubic Bézier triangles on the face mesh refined with two levels of Loop subdivision. To create the perspective displacement map, we apply the displacement along the view rays $\frac{\bar{\mathbf{p}}_{ij}}{\|\bar{\mathbf{p}}_{ij}\|_2}$. Therefore, the new

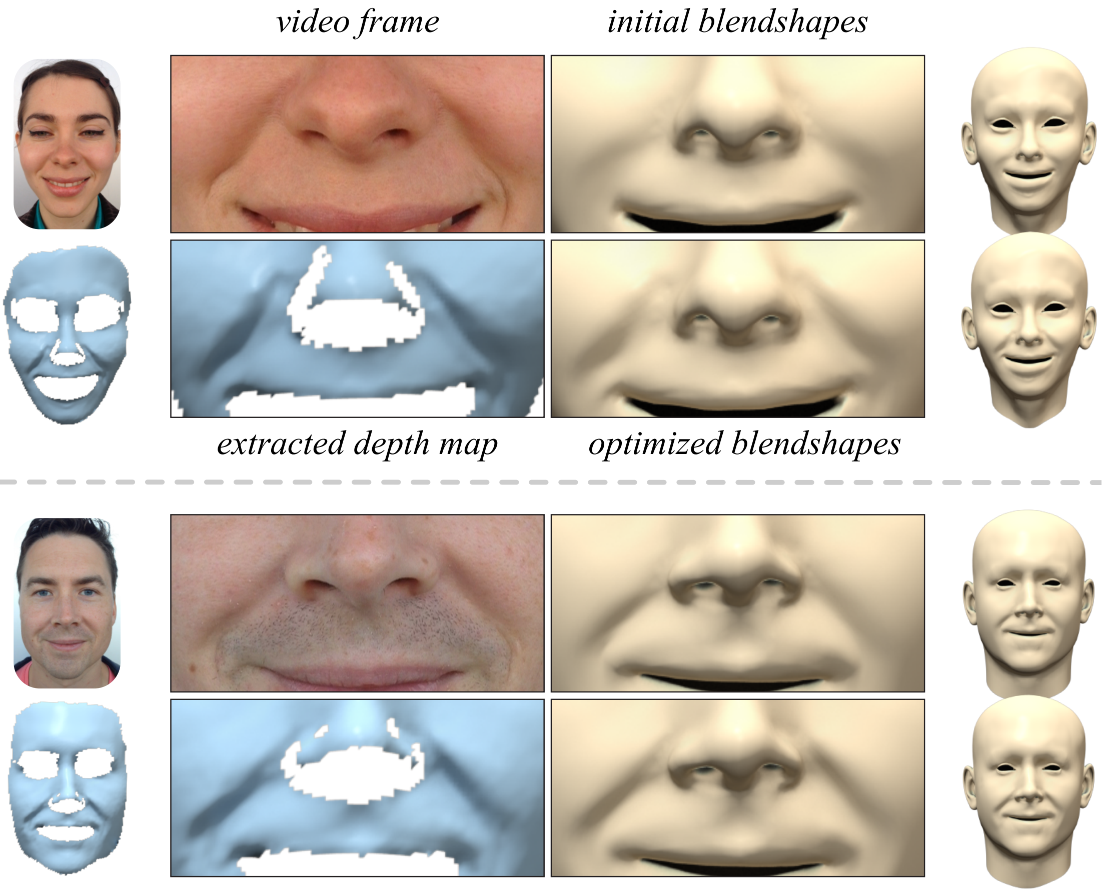


Figure 2.11 – Geometry refinement adds further nuances to the reconstructed blendshapes. For each frame of the video sequence we extract a depth map using shape-from-shading that serves as a constraint for the refinement optimization.

3D point location $\mathbf{p}_{ij}$ of the pixel $\{i, j\}$ can be expressed as

$$\mathbf{p}_{ij} = \bar{\mathbf{p}}_{ij} + d_{ij} \frac{\bar{\mathbf{p}}_{ij}}{\|\bar{\mathbf{p}}_{ij}\|_2}, \tag{2.11}$$

where d_{ij} is the displacement value for this pixel. The normal at that pixel can then be estimated as

$$\mathbf{n}_{ij} = \frac{1}{N_{ij}} (\mathbf{p}_{i+1,j} - \mathbf{p}_{ij}) \times (\mathbf{p}_{i,j+1} - \mathbf{p}_{ij}), \tag{2.12}$$

where $N_{ij} = \|(\mathbf{p}_{i+1,j} - \mathbf{p}_{ij}) \times (\mathbf{p}_{i,j+1} - \mathbf{p}_{ij})\|_2$. Let $\mathbf{d}$ be a vector that stacks all the displacements d_{ij} and $\mathbf{y}$ be the vector of spherical harmonics coefficients. To reconstruct the displacement map we optimize

$$\min_{\mathbf{d},\mathbf{y}} \sum_{ij} \left\| \begin{bmatrix} \rho_{i+1,j}\mathbf{s}_{i+1,j} - \rho_{ij}\mathbf{s}_{ij} \\ \rho_{i,j+1}\mathbf{s}_{i,j+1} - \rho_{ij}\mathbf{s}_{ij} \end{bmatrix} - \begin{bmatrix} c_{i+1,j} - c_{ij} \\ c_{i,j+1} - c_{ij} \end{bmatrix} \right\|_2^2 + \\ + \mu_1 \|\mathbf{d}\|_2^2 + \mu_2 \|\mathbf{Gd}\|_2^2 + \mu_3 \|\mathbf{Ld}\|_2^2 \tag{2.13}$$

over $\mathbf{d}$ and $\mathbf{y}$, where c_{ij} is the grayscale value at pixel $\{i, j\}$ of the input frame and $\mathbf{s}_{ij} = \mathbf{y}^T \phi(\mathbf{n}_{ij})$. Similar to Equation 2.3, we regularize the displacements to be smooth and of low magnitude. To solve this optimization we alternately minimize Equation 2.13 over $\mathbf{y}$ by solving a linear system with fixed normals initialized from the original mesh, and over $\mathbf{d}$ with fixed weights $\mathbf{y}$ using a Gauss-Newton method. The depth and normal maps are then computed from the displacement maps using Equation 2.11 and Equation 2.12, respectively.

After extracting the depth and normal maps, we use a non-rigid registration approach to refine the blendshapes. We formulate a registration energy

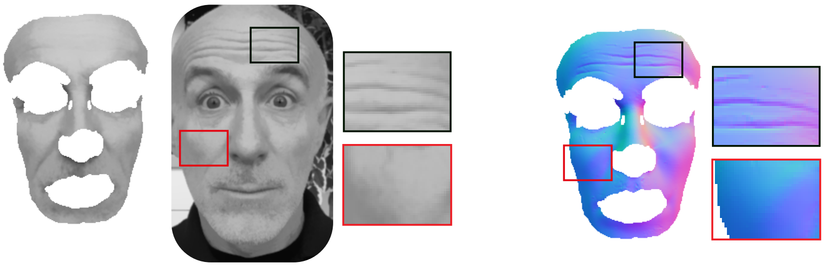
$$E_{\text{reg}} = \sum_{v \in \mathcal{V}} \|\mathbf{n}_v^T (\mathbf{F}_\mathbf{T}(\mathbf{B}_v, \mathbf{w}) - \mathbf{p}_v)\|_2^2, \tag{2.14}$$

where $\mathcal{V}$ is the set of blendshape vertices, $\mathbf{p}_v$ is the closest point of $\mathbf{F}_\mathbf{T}(\mathbf{B}_v, \mathbf{w})$ on the depth map, and $\mathbf{n}_v$ is the normal at that point. This energy is optimized over the blendshapes $\mathbf{B}$ jointly over the whole sequence combined with a closeness and a smoothness energy (Equation 2.9 and Equation 2.10, respectively).

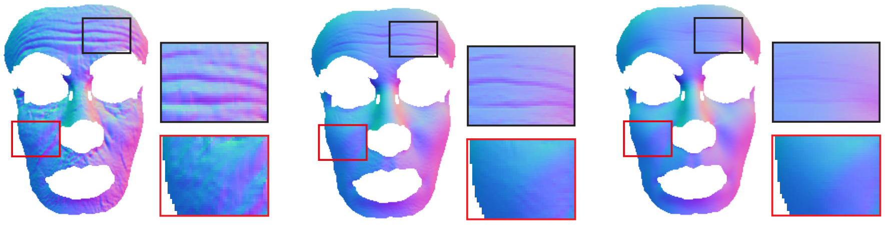
2.5.2 Reconstructing Detail Maps

In high-end studio systems, fine-scale details such as wrinkles are commonly directly encoded into the mesh geometry [BBB*10, GVWT13]. However, this requires a very fine discretization of the mesh which may not be suitable for realtime animation and display. Instead, we create a set of detail maps in an offline optimization to enable realtime detail synthesis and rendering at animation runtime.

Similar to the geometric refinement step, we extract one depth map per frame. This time the input video is downsampled 4 times to a resolution of 270 × 480 in order to keep small-scale details while reducing noise. To reconstruct sharp features we modify Equation 2.13 by replacing the ℓ_2 norm in the smoothness energies by an ℓ_1 norm. The ℓ_1 norm has been widely employed for image processing tasks such as denoising [CCC*10] as it allows preserving sharp discontinuities in images while removing noise. To solve the ℓ_1 optimization, Gauss-Newton is adapted using an iterative reweighing approach [CY08]. The normal maps are then computed



albedo *video frame* *L1 norm*



L2 norm, 1e7 weight *L2 norm, 1e9 weight* *L2 norm, 1e10 weight*

Figure 2.12 – Our reconstruction of detail maps uses ℓ_1-norm optimization to separate salient features such as the wrinkles on the forehead from noise. The lower row shows that ℓ_2 optimization with a low smoothness weight retains too much noise (left), while increasing the smoothness weights blurs out salient features (right).

from the displacement maps using Equation 2.12. Figure 2.12 shows a visualization of the effect of the ℓ_1-norm in the extraction of the detail maps.

After extracting normals, we compute ambient occlusion maps by adapting the disk based approach proposed in [Bun05] to texture space, where we directly estimate ambient occlusion coefficients from the extracted normal and displacement maps. For each pixel p we calculate the ambient occlusion value $\mathrm{ao}(p)$ by sampling a set $\mathscr{S}_p$ of nearby pixels such that

$$\mathrm{ao}(p) = 1 - \sum_{k \in \mathscr{S}_p} \left(1 - \frac{1}{\sqrt{\frac{1}{\|\mathbf{v}_{pk}\|_2} + 1}}\right) \frac{\sigma(\mathbf{v}_{pk}^T \mathbf{n}_p)\sigma(\mathbf{v}_{pk}^T \mathbf{n}_k)}{|\mathscr{S}_p|}, \tag{2.15}$$

where $\sigma(x)$ clamps x between 0 and 1, $\mathbf{n}_k$ is the normal at pixel k of the normal map, and $\mathbf{v}_{pk}$ is the vector between the 3D locations of pixels p and k reconstructed using the displacement maps.

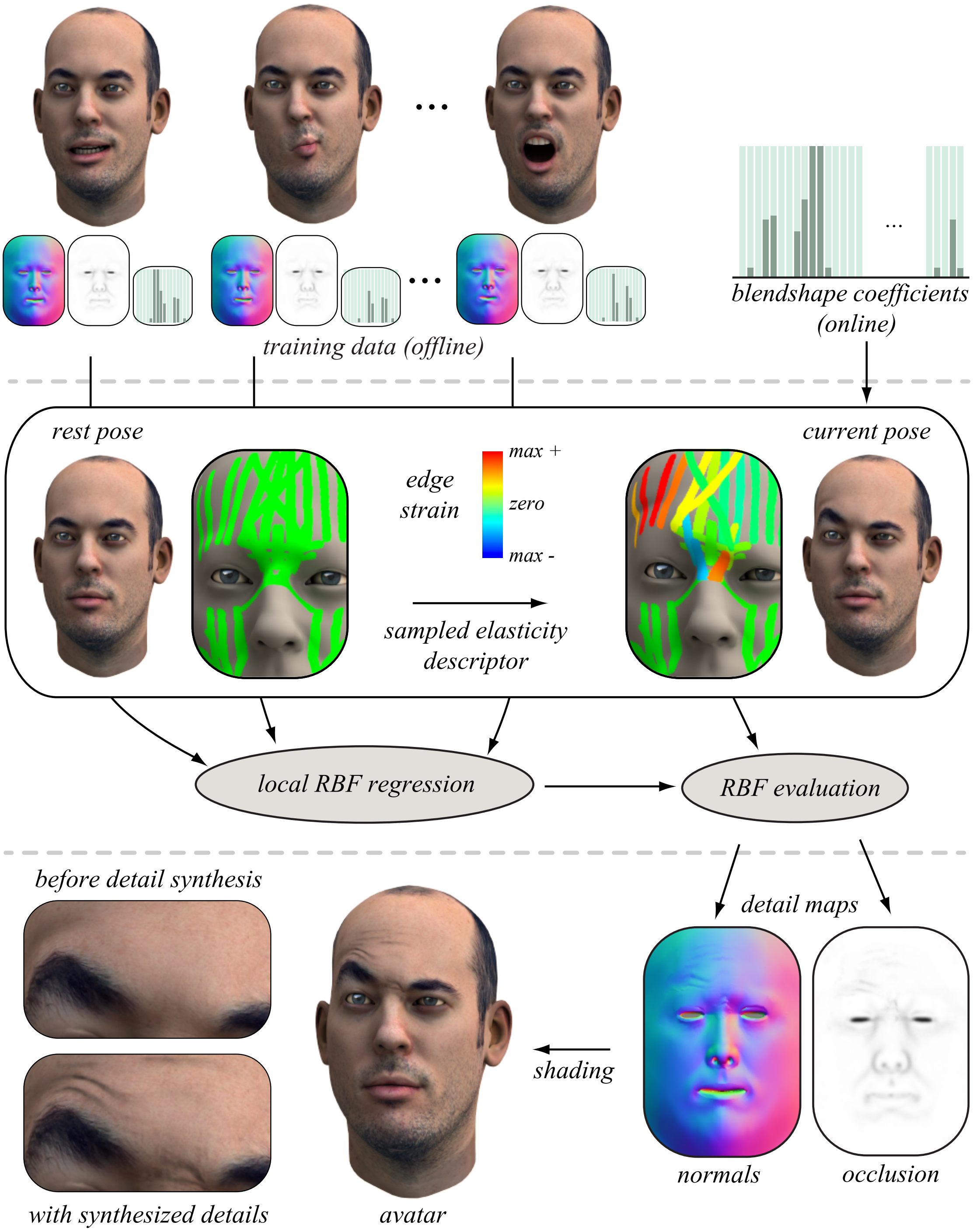


Figure 2.13 – On-the-fly detail synthesis. Blendshape coefficients drive the reconstructed face rig during realtime animation. For each animated expression, pose-specific details are synthesized using an RBF regressor that is trained with the detail maps reconstructed offline during dynamic modeling. The RBF function is evaluated based on deformation strains measured on a sparse set of edges (colored).

2.6 Animation

The dynamic reconstruction stage provides us with a user-adapted blendshape model and a set of high-resolution detail maps containing normal and ambient occlusion maps that correspond to the recorded expressions of the video sequence. The blendshape representation allows for simple and intuitive animation. Blendshape coefficients can be directly mapped to animation controllers for keyframe animation or retargeted from face tracking systems (see also Figure 2.19). To augment the blendshape rig, we synthesize dynamic details on the fly by blending the reconstructed detail maps of the dynamic modeling stage using a local strain measure evaluated on the posed blendshape meshes (see Figure 2.13).

Detail Map Regression. Our detail synthesis method is inspired by the approach of [BLB*08] that links edge strain to a displacement function. In contrast, we learn a mapping between edge strain and normal and ambient occlusion maps which facilitates more efficient detail synthesis using GPU shaders.

In a preprocessing stage, we train a radial basis function (RBF) regressor using the detail maps extracted for each frame of the tracked sequences and a strain measure computed on a sparse set of feature edges $\mathcal{E}$ defined on the mesh (see Figure 2.13). We compute the strain value of an edge $e \in \mathcal{E}$ as $f_e = (\|\mathbf{e}_1 - \mathbf{e}_2\|_2 - l_e)/l_e$, where $\mathbf{e}_1$ and $\mathbf{e}_2$ are the positions of the edge endpoints and l_e is the edge rest length. We then learn the coefficients w of an RBF regressor independently for each layer of the detail map. The regression for each pixel $\{i, j\}$ of a particular layer is formulated as

$$\mathbf{l}_{ij}(\mathbf{f}) = \sum_{k\in\mathcal{K}} \eta_k \varphi\left(\|\mathbf{D}^{\frac{1}{2}}_{ij,k}(\mathbf{f}-\mathbf{f}_k)\|_2\right), \tag{2.16}$$

where $\mathcal{K}$ is a set of selected keyframes, $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_k]$ are the RBF weights, and $\mathbf{f} = [f_1, \ldots, f_{|\mathcal{E}|}]^T$ is a vector stacking the strain f of all feature edges. We employ the biharmonic RBF kernel $\varphi(x) = x$ in our implementation. To localize the strain measure, we integrate for each keyframe a per-pixel diagonal matrix $\mathbf{D}_{ij,k} = \text{diag}(\alpha_{ij,k,1}, \ldots, \alpha_{ij,k,|\mathcal{E}|})$. Dropping the index $_{ij,k}$ for notational brevity, we define the weight α_e for each edge $e \in \mathcal{E}$ based on the distance of the pixel $\{i, j\}$ with 3D position $\mathbf{p}_{ij}$ to the edge e as

$$\alpha_e = \frac{\bar{\alpha}_e}{\sum_{l\in\mathcal{E}} \bar{\alpha}_l} \quad \text{with} \quad \bar{\alpha}_e = \exp^{-\beta(\|\mathbf{p}_{ij}-\mathbf{e}_1\|_2+\|\mathbf{p}_{ij}-\mathbf{e}_2\|_2-l_e)}. \tag{2.17}$$

The parameter β controls the drop-off. The localization spatially decouples the keyframes to avoid global dependencies and facilitates independent detail synthesis for different regions of the face. The RBF weights **w** are trained by minimizing the reconstruction error to the frames of the tracked sequences. The keyframes are selected greedily by sequentially adding the frame with maximum reconstruction error.

Detail Synthesis. The trained RBF regressor can now be used for detail synthesis during animation. The face rig is driven by blendshape coefficients. For the posed mesh, we compute the strain vector of the feature edges and evaluate Equation 2.16 to create new detail maps. The synthesized normal and ambient occlusion maps are then applied in the pixel shader.

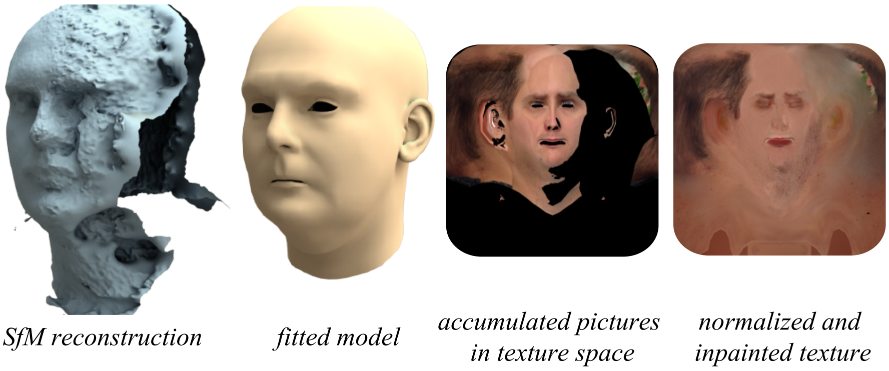
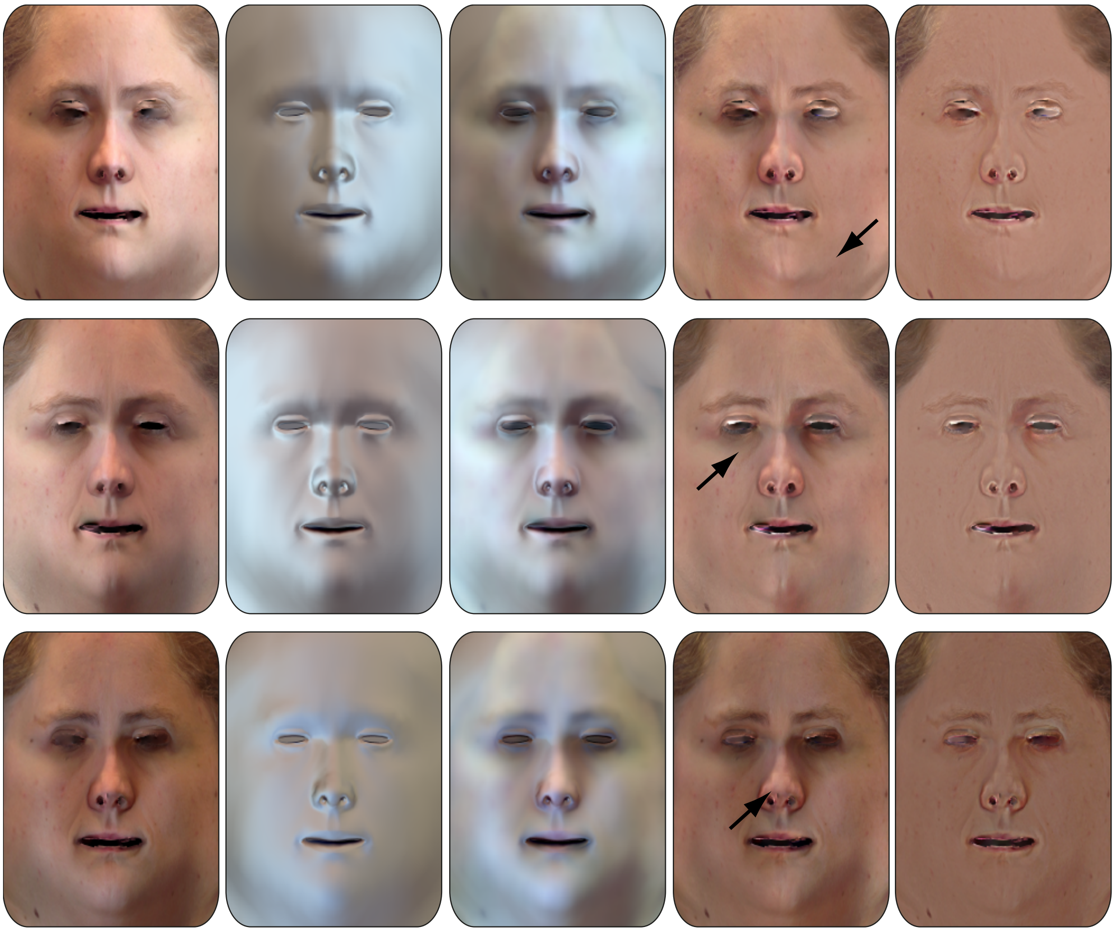

SfM reconstruction *fitted model* *accumulated pictures in texture space* *normalized and inpainted texture*

Figure 2.14 – Degrading input data, in this case missing images on the cheek of the captured user, can lead to lower accuracy in the reconstructed static pose and texture artifacts produced by the inpainting algorithm (c.f. with Figure 2.5).

2.7 Evaluation

We applied our dynamic 3D avatar reconstruction pipeline on a variety of subjects as shown in Figures 2.1 and 2.17. For all subjects, we use around 80 images for the static reconstruction and less than 90 seconds of video for the dynamic modeling. These examples illustrate that our approach faithfully recovers the main geometric and texture features of the scanned subjects. We also show the effect of on-the-fly detail synthesis. The combination of per-pixel normals and ambient occlusion coefficients, which can both be integrated efficiently into per-pixel shading models, leads to further improvements on the appearance of the animated face rig (see also accompanying video).

Data Quality. We investigated how the output of our algorithm is affected by degrading input data quality. In particular, insufficient coverage of the face for the acquisition of the static model can lead to artifacts in the reconstruction. This lack of coverage can either result from



Poisson integrated

Spherical harmonics

Spherical harmonics + corrective field

Albedo (spherical harmonics)

Albedo (spherical harmonics + corrective field)

Figure 2.15 – Our lighting factorization approach successfully normalizes the light in three datasets captured under different illuminations. Notice how the corrective field aids in capturing shadows and specularities better than using spherical harmonics alone.

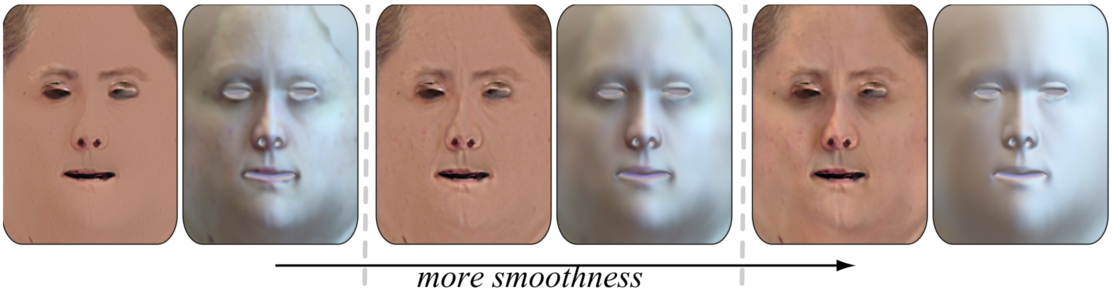


Figure 2.16 – The influence of the parameters in the albedo extraction. By increasing the smoothness of the corrective field using the values of λ_2 and λ_3, more details are captured in the albedo.

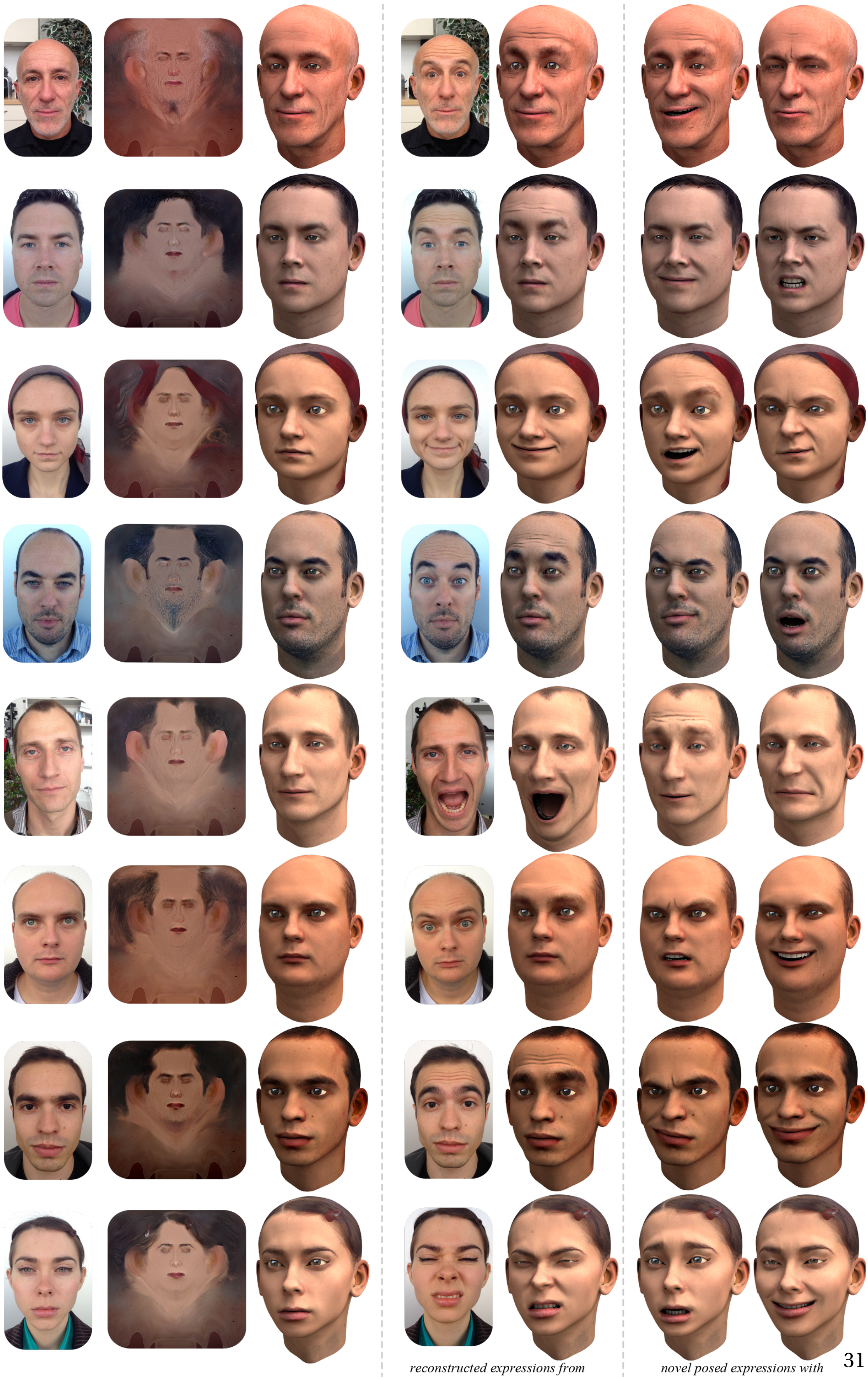

input image
albedo texture
neutral expression
reconstructed expressions from the input video with detail maps
novel posed expressions with dynamically synthesized detail maps

Figure 2.17 – Fully rigged 3D facial avatars of different subjects reconstructed with our method.

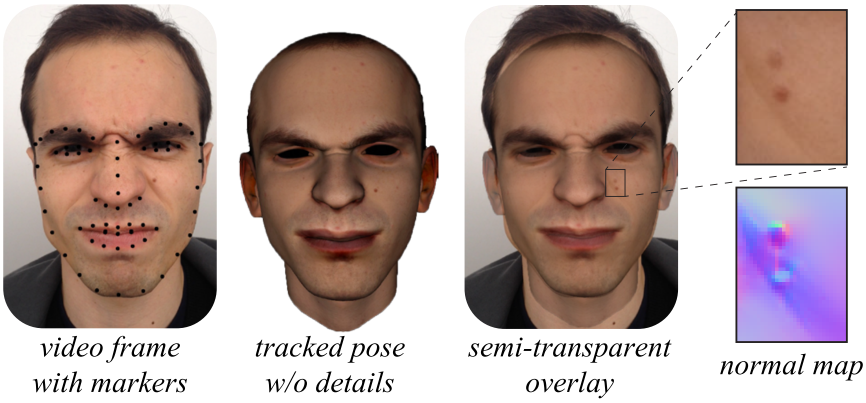


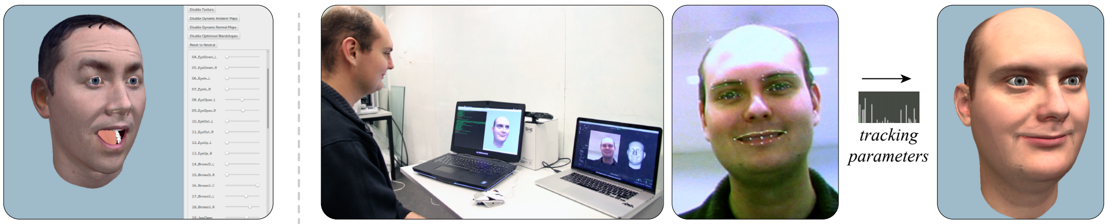
Figure 2.18 – Misalignments between video frames and the textured model can cause inconsistencies in the detail maps, here most visible at the mole on the cheek.

the user failing to capture sufficiently many images, or from images being automatically discarded by the MVS algorithm due to, for example, excessive motion blur. Figure 2.14 illustrates how artifacts in the reconstructed point cloud can to a certain extent be compensated by the PCA regularization at the cost of diminished resemblance to the recorded user. Similarly, texture inpainting can fill missing texture information, but leads to visible artifacts in the reconstruction. While more sophisticated inpainting methods could alleviate these artifacts, we found that the simplest solution is to give visual feedback to the user to ensure adequate data capture. In all our experiments, users were able to record images of sufficient quality after being instructed about potential pitfalls in the acquisition, such as fast camera motion, changing illumination during capture, or insufficient coverage of the face.

Light Extraction. We ran experiments to verify the robustness of our albedo extraction given different lighting conditions, using the same set of parameters. Figure 2.15 displays the results for one of our test subjects, as well as the intermediate lighting estimations. Due to ambiguity introduced by the dependency of the captured skin color to the real skin color and the environment lighting, considering the skin color as the median color inside the face mask outputs slightly different results.

Furthermore, Figure 2.16 shows the behaviour of the albedo extraction under different parameters. We vary the smoothness of the corrective field in equation 2.3, regularizing the level of detail included into the extracted lighting.

Texture alignment. In general, the combination of feature energy and optical flow constraints in the tracking optimization of Equation 2.4 yields accurate alignments between the textured model and the video frames. However, in areas far away from the tracked features, such as the cheek, texture misalignments can occur that in turn can lead to reconstruction errors in the detail maps (see Figure 2.18). A possible solution to this problem is to adapt the optical flow energy of Equation 2.6 to incorporate additional texture features computed, for example, using SIFT descriptors.



interactive character posing *Faceshift Studio live tracking* *realtime animation*

Figure 2.19 – Application demos utilizing our rigged avatars. Left: an interactive tool for posing the avatar by directly controlling blendshape weights. Right: The avatar is animated in realtime by streaming blendshape coefficients from a realtime tracking software.

Limitations. The simplicity of our acquisition setup implies a number of limitations in terms of scanning accuracy. As indicated above, limited spatial and temporal resolution of the camera, sensor noise, motion blur, or potentially insufficient illumination can adversely affect the reconstruction results.

Our albedo factorization works well in casual lighting scenarios, but cannot fully handle high specularity or hard shadows in the acquired images. For such adverse lighting conditions, artifacts in the reconstructed albedo are likely to occur.

Blendshape models also have some inherent limitations. In particular, unnatural poses can be created for extreme expressions such as mouth very wide open, since a proper rotation of the lower lip is not represented in the linear model. Popular remedies, such as corrective shapes or a combination with joint-based rigging could potentially be integrated into our system, at the expense of a more complex tracking optimization.

A further limitation of our method is that we do not represent nor capture hair. This means that currently we can only reconstruct complete avatars for subjects with no hair or where the hair can be appropriately represented as a texture. More complex hair styles need to be treated separately outside our pipeline. Recent progress on hair capture [HMLL14] and realtime hair animation [CZZ14] offer promising starting points to further investigate this challenging problem. We also do not capture the teeth or tongue, but simply scale the template geometry appropriately.

Applications. Figure 2.19 shows reconstructed avatars in two application scenarios. Interactive character posing for keyframe animation is facilitated through direct control of the blendshape weights. Please see the additional material for a demo application that allows animating a reconstructed character in realtime. Alternatively, the character can be animated by transferring blendshape weights from a face tracking application. We use the commercial tool faceshift Studio that allows realtime streaming of blendshape coefficients. This demo illustrates the potential of our approach to bring personalized 3D avatars into consumer-level applications.

Future Work. Beyond addressing the limitations discussed above, we identify several interesting avenues for future work. Recent advances in RGB-D cameras show great promise of bringing active depth sensing into mobile devices such as tablets or phones. This opens up interesting possibilities for new reconstruction algorithms that directly exploit the acquired depth maps.

Integrating sound seems a promising extension of our method, both on the reconstruction and the synthesis side. For example, an analysis of recorded speech sequences could guide the tracking and reconstruction of the blendshape model and detail maps. Avatars could also be driven by text-to-speech synthesis algorithms.

The possibility to transfer detail maps between subjects (see Figure 2.20) not only allows modifying the reconstructed avatars, but can potentially also simplify the acquisition process. Statistical priors for wrinkle formation could be learned from examples, given a sufficiently large database.

Further research is also required to answer important questions related to the perception of virtual avatars, such as: How well does an avatar resemble the user? How well does an animated avatar convey the true emotions of a tracked user? or What reactions does the virtual model evoke in online communication? We believe that these questions, along with the ultimate goal of creating complete, video-realistic 3D avatars with a consumer acquisition system lays out an exciting research agenda for years to come.

2.8 Conclusion

We have introduced a complete pipeline for reconstructing 3D face rigs from uncalibrated hand-held video input. While this minimalistic acquisition setup brings the creation of personalized 3D avatars into the realm of consumer applications, the limited input data quality also poses significant challenges for the generation of consistent and faithful avatars. Our

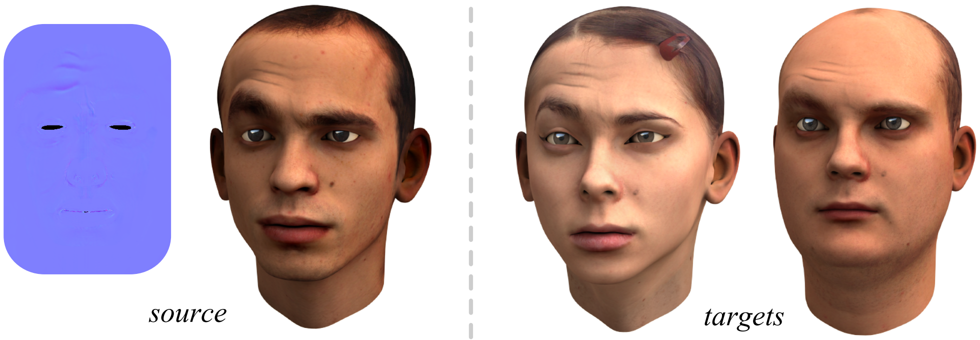


Figure 2.20 – Detail maps can easily be transferred between subjects thanks to the consistent parameterization of the blendshape meshes across all avatars.

solution combines carefully designed reconstruction priors, a two-scale dynamic blendshape representation, and advanced tracking and reconstruction algorithms to minimize the required user assistance while maximizing reconstruction quality. We believe that our solution provides an important first step towards realtime avatar-based interactions for the masses, which could have a significant impact on the way we communicate in virtual worlds.

2.9 Implementation Details

Our software is implemented in C++ and parallelized on the CPU using OpenMP. We use the Eigen library for fast linear algebra computations and OpenCV for all the image processing operations. Our implementation runs on a laptop with an Intel Core i7 2.7Ghz processor, 16 GBytes of main memory, and an NVIDIA GeForce GT 650M 1024MB graphics card.

Static Modeling. The dense point cloud reconstruction with about 500k points takes 30 to 40 minutes for approximatively 80 pictures. The static modeling is then performed using the identity PCA model of [BV99]. We use 50 PCA basis vectors to approximate the neutral expression. The registration problem is optimized with Gauss-Newton using the supernodal Cholmod sparse Cholesky factorization. The non-rigid registration takes approximatively 10 seconds.

For the static model, we generate a high-resolution albedo texture of 4096 × 4096 pixels. To efficiently solve the Poisson integration [PGB03] and to minimize Equation 2.3 over the corrective fields we use the Matlab Engine FFT. The parameters of Equation 2.3 are set to $\lambda_1 = 10^2$, $\lambda_2 = 10^{-3}$, and $\lambda_3 = 10^3$ for all the examples. The static texture is created in approximatively 5 minutes.

Dynamic Modeling. In our current implementation we employ a blendshape model of 48 blendshapes (see also accompanying material). The input videos are recorded at 30Hz with an average length of 1 minute. The videos are temporally downsampled to 3Hz prior to processing. We then apply a multiresolution approach with a four-level image pyramid. The optimization is first solved on the coarsest level, the solution is then propagated as an initialization to the next finer level until reaching the original resolution. The combined tracking and modeling optimization takes approximatively 60 seconds per frame. We perform the tracking optimization using a warm started shooting method [Fu98], and the modeling using Gauss-Newton.

The parameters are set to $\gamma_1 = 10^{-1}$, $\gamma_2 = 10^{-2}$, $\gamma_3 = 10^4$, $\gamma_4 = 10^4$, $\gamma_5 = 10^2$, and $\gamma_6 = 10^8$ for all our examples.

To solve the shape-from-shading optimization we use Gauss-Newton. Symbolic sparse Cholesky factorization is used to improve performance as the sparsity pattern of the system matrix remains constant. Computation time is around 5 seconds for extracting a 150 × 240 depth map for the geometric refinement. The detail map extraction takes 25 seconds for a 1024 × 1024 normal map and another 5 seconds for the corresponding ambient occlusion map. The optimization weights are set to $\mu_1 = 10^6$, $\mu_2 = 10^3$, and $\mu_3 = 10^7$ for the geometric refinement, and $\mu_1 = 10^6$, $\mu_2 = 10^4$, and $\mu_3 = 10^6$ for the detail map extraction. The non-rigid refinement of the blendshape model is performed in about 60 seconds. The parameters are set to $\gamma_4 = 10^5$, $\gamma_5 = 1$, and $\gamma_6 = 10$.

Animation. We implemented the RBF evaluation on the GPU using a GLSL fragment shader. For 6 keyframes of size 1024 × 1024, and 44 strain edges the animation can be performed at realtime frame rates, i.e., 100 fps. The training of the RBF regressor takes approximatively 5 seconds. The parameter β is set to 150 for our meshes with an average edge length of 4.1 cm.

Manual User Interaction. From our trials, we concluded that about 15 minutes are needed to perform the manual feature corrections necessary for the static ($\sim 1-2$ minutes) and dynamic reconstructions ($\sim 7-15$ minutes). The additional video shows a complete example. The decision of using [SLC09] was based on code and documentation availability, and we believe that more recent and precise methods such as [CWLZ13, CWWS12] could be used to reduce or eliminate the amount of manual corrections.

Retrospective

As mentioned already in Section 2.5, the feature extraction algorithm used in our implementation is that of Saragih et al. [SLC09] and it was chosen back then for its mature open-source implementation. In the meantime, multiple works have been published with more advanced techniques for facial feature extraction [LKA*17]. Not only would these potentially improve the tracking accuracy and performance of our pipeline, but they would significantly reduce the need for the user assistance in correcting the landmark locations.

In this paper we have extensively employed the thin-shell deformation model as a regularizer for whenever we were deforming the shape of the face outside the PCA or template blendshape space. In one of our subsequent publications [IKNDP16] we have proposed an anatomically-inspired volumetric model that fixes some of the artifacts present in surface-only regularization models. In particular, this helps in preserving the volume of the face when performing expressions as well as helps avoid self-intersections of the mesh by explicitly resolving collisions. For a more detailed presentation, we refer the reader to Chapter 5.

Along the lines of departing from blendshape models, worth mentioning are the works of Garrido et al. [GZW*16] and Wu et al. [WBGB16]. The first explores corrective blendshapes for improving the tracking of lips in monocular videos. The second work uses localized face patches together with anatomically-inspired constraints in order to extract per-frame geometry from monocular videos. Similarly, Cao et al. [CBZB15] use trained local regressor to add medium- and small-scale facial details such as wrinkles to a tracked performance sequence. None of these methods aim at creating a consistent facial avatar, but rather extract good quality geometry for each frame of the input video. It would be an interesting venue of work to explore how to build a consistent avatar from such results.

Our work does not capture hair, and until now there have not been convincing approaches to capturing hair using data from consumer-level devices [HBLB17]. However, techniques for reconstructing personalized teeth models have been proposed [WBG*16a], as well as for inferring realistic eyes [BBGB16].

In order to improve the robustness to missing areas or low quality input data, a work that would be interesting to integrate within our framework is that of Saito et al.[SWH*16], which uses deep neural networks to infer high resolution face textures given a single low quality portrait.

Furthermore, regarding the deployment of such software, smart phone capabilities have progressed since the publication of this paper. Notably, phones with dual cameras or fullfledged RGB-D sensors are becoming available, such as the ones with the Google Tango technology [Goo17]. We expect such devices to become more widespread in the next years as AR applications get more and more mature. Having a good quality depth map together with color information would simplify some of the tracking and geometric steps significantly.

Later in this thesis (Chapter 6) we will look into a completely different approach to animating face avatars. Instead of relying on deformable blendshape models, we shall explore anatomically-motivated techniques for actuating muscle regions in order to deform the skin of the digital actor to the desired shape. Moreover, we will show in the results section that the facial geometry created using the approach presented in this chapter can be used directly as targets in our newer work relying on physics simulation.

3 Semantic Parametric Body Shape Estimation from Noisy RGB-D Sequences

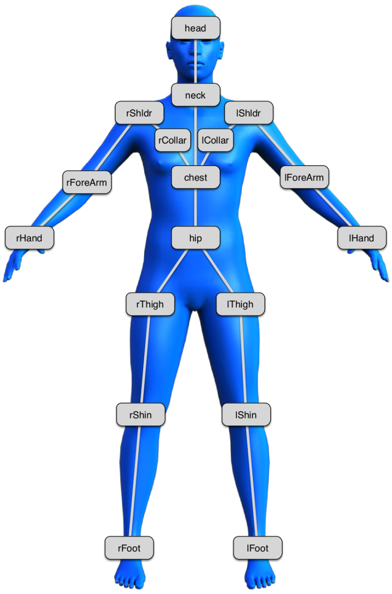


Figure 3.1 – MakeHuman parametric body model with an overlaid example skeleton.

Note

This chapter is based on the following publication [IT16]:

ICHIM, A.E. AND TOMBARI, F. Semantic Parametric Body Shape Estimation from Noisy RGB-D Sequences. *Robotics and Autonomous Systems, 2015*

The candidate contributed with most of the scientific contributions and implementation of this publication.

Abstract

The paper proposes a complete framework for tracking and modeling articulated human bodies from sequences of range maps acquired from off-the-shelf depth cameras. In particular, we propose an original approach for fitting a pre-defined parametric shape model to depth data by exploiting the 3D body pose tracked through a sequence of range maps. To this goal, we make use of multiple types of constraints and cues embedded into a unique cost function, which is then efficiently minimized. Our framework is able to yield compact semantic tags associated to the estimated body shape by leveraging on semantic body modeling from MakeHuman and L1 relaxation, and relies on the tools and algorithms provided by the open source Point Cloud Library (PCL), representing a good integration of the functionalities available therein.

3.1 Introduction and Related Work

This paper is accompanied by a webpage containing the datasets and source code for the approach presented in the paper, as well as other samples: http://lgg.epfl.ch/~ichim/bodies_ras_2015/.

The task of 3D body modeling aims at automatically obtaining an accurate 3D model of a person's body. The possibility of having at disposal an accurate 3D model adapted to the body characteristics of a subject opens up new directions in a variety of applications, such as in the fields of entertainment (e.g. 3D avatar creation for videogaming and movie special effects), fitness (e.g., for automatic estimation of the body mass), apparel (e.g., for virtual changing room applications), interactive design, and security (people detection and identification).

The output of this task is generally represented by a parametric 3D body model, with the parameters estimated so that the model adapts to the specific characteristics of the subject being scanned. It is often the case that these parametric models are open sourced and available to the community so to favor interchange and standardization. While earlier parametric models [ACP03] were based on simple Principal Component Analysis (PCA) of standard human poses (e.g., T/A poses), more recent approaches also model minute body deformations such as muscle bulging under complex poses, e.g., the SCAPE models [ASK*05, HSS*09]. Another possibility of sourcing parametric body models is from semantic models, i.e., models built by artists, where each body shape modifier has an associated semantic tag, such as it is the case of MakeHuman [BRM08].

Accurately estimating the 3D body model traditionally requires dedicated and expensive hardware to acquire high resolution scans of the body, generally by means of 3D laser scanners or high frame-rate structured light sensors. In addition, this procedure is characterized by high processing time due to the re-positioning of the scanner from different view points, the acquisition and the joint 3D registration of the different scans. To overcome such limitations, the work of [GWBB09] proposed to fit a 3D parametric model to a frame acquired by means of a monocular RGB camera. Although not fully automatic due to the need of user interaction as well as limited in the modeling accuracy due to the 2D to 3D fitting, this work introduced the concept of using low-cost hardware for the task of 3D body modeling.

Successively, thanks to the popularity of consumer depth cameras originated by the development of the Microsoft Kinect, other works [WHB11, MKHG13, HBB*13, YY14] have tackled 3D body modeling by means of the noisy range data acquired from such low-cost 3D sensors. Initially, [WHB11] proposed to fit each parametric 3D model obtained from SCAPE [ASK*05] on a certain number of range depth maps (e.g. 4) by optimizing an objective cost fuction relying on 3D data fitting as well as silhouette fitting. The main limitations of such a method are represented by the constraints imposed by the system, in the form of a specific pose (*T-pose*) that the subject has to assume throughout the sequence, and by the overall efficiency (more than one hour is reported to process one subject). Successively, in [HBB*13], simplified SCAPE shape models are estimated from two depth maps of the subject (one frontal, one from the back) in real time by optimizing a cost function composed of two terms, respectively taking into account point-to-point and point-to-plane fitting. Analogously to [WHB11], this method carries out the modeling by relying on a small number of slightly overlapping frames, hence might suffer from the presence of noise in the data.

Differently, non-parametric shape modeling approaches have been also proposed. This is the case of [MKHG13], where a moving voxel grid is used for each body part to integrate together surface measurements obtained from a depth map sequence within a Truncated Signed Distance Function (TSDF) representation, this allowing to build volumetric models for both the background and each piecewise body part. Due to the TSDF fusion, the output is not a parametric body model, but a piecewise smooth 3D mesh reconstruction of the body. Another non-parametric approach is the one proposed in [YY14], where real-time pose and shape estimation is obtained via a probabilistic approach based on a Gaussian Mixture Model (GMM). Also in this case, the input is represented by a sequence of RGB-D frames. A purely point-based technique is proposed by [MBF*14] for the people re-identification task; the authors use the Microsoft SDK to track and segment the body, and then the points are accumulated by transforming each limb to the standard A-pose.

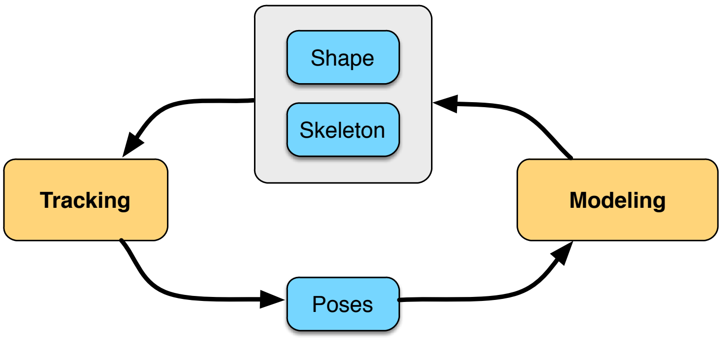


Figure 3.2 – Overview of the proposed tracking and modeling pipeline

In this work, we propose a framework aimed at efficient 3D parametric body modeling from noisy depth sequences acquired with consumer depth cameras. Conversely to [WHB11, HBB*13], one main contribution of this work is to leverage on the temporal cue by explicitly tracking the 3D subject and estimating its 3D pose through a sequence of frames. This allows us to integrate the noisy body shape of the subject over many temporally correlated frames, effectively averaging out noise. The modeling procedure is carried out by minimization of an energy cost which includes, as a second contribution of our approach, additional set of cues with respect to those used in previous works, based on silhouette and 3D surface fitting, as well as skeleton similarity, PCA and smoothness. We show that the combinations of these terms induces a more robust estimation of the body model. Finally, and differently to [WHB11, HBB*13], we propose to use MakeHuman models [BRM08] due to their better integration with semantic information associated to each body part. In conjuction with this, a third contribution is a specific L1 minimization of the energy cost term associated with our modeling scheme, so to induce sparsity in the semantic tags and automatically yield a compact semantic description of each acquired body.

In Sections 3.2 and 3.3 we illustrate the entire proposed pipeline, which tracks the body motion of the subject and estimates the pose of its body joints over time, then, from the 3D estimated body pose at each frame, it refines the parameters of a MakeHuman body model by cost function optimization. A graphical overview of the proposed pipeline is shown in Figure 3.2. Our framework relies on open-source computer vision and full body modeling libraries such as the Point Cloud Library (PCL) and MakeHuman, and it is easily customizable for different tasks requiring different precision and performance due to the modularity of its nature. In our initial implementation it is able to process frames at a speed of 3 fps and does not require specific constraints on the pose of the subject.

To demonstrate the effectiveness of our approach, in Section 3.4 we show some qualitative examples of body models estimated and tracked from real data acquired from consumer depth cameras, as well as measured accuracy of the estimated body model with respect to specific body parts. We also demonstrate the usefulness of compact semantic body tags associated to our estimated body models.

3.2 Proposed methodology

3.2.1 Data Representation

In our system, the articulated human bodies are represented as quad and/or triangle meshes. The bodies can be articulated via the underlying skeleton. Skeletons are composed of multiple joints disposed in a tree hierarchy (see Figure 3.1 for an example), based on which local node transformations are propagated: $\boldsymbol{T}^{j}_{abs} = \boldsymbol{T}^{parent(j)}_{abs}\boldsymbol{T}^{j}_{local}$, where $\boldsymbol{T}^{j}_{abs}$ is the world transformation of joint j, and $\boldsymbol{T}^{j}_{local}$ is its local transformation with respect to its parent node in the skeleton tree. Each joint influences a number of mesh vertices in its vicinity, as defined by the linear blend skinning model: $\boldsymbol{v}^{pose}_{i} = \sum_j w^{j}_{i}\boldsymbol{T}^{pose}_{j} * (\boldsymbol{T}^{rest}_{j})^{-1} * \boldsymbol{v}^{rest}_{i}$, where each joint j in the skeleton has $\boldsymbol{T}^{rest}_{j}$ as the transformation corresponding to the rest pose (A or T-pose) and $\boldsymbol{T}^{pose}_{j}$ the transformation of the joint in the posed skeleton configuration, and w^{j}_{i} is the blend skinning weight of joint j over mesh vertex $\boldsymbol{v}_i$. The joints are modelled by their rest transformation (expressed using rotation matrix $\boldsymbol{R}^{rest}_{j}$ and translation vector $\boldsymbol{t}^{rest}_{j}$) and the pose rotation parametrized using Euler angles $\boldsymbol{\beta}$: $\boldsymbol{T}^{pose}_{j} = \boldsymbol{R}^{rest}_{j} * \boldsymbol{R}^{x}(\beta^{x}_{j}) * \boldsymbol{R}^{y}(\beta^{y}_{j}) * \boldsymbol{R}^{z}(\beta^{z}_{j}) + \boldsymbol{t}^{rest}_{j}$.

The skeleton model deforms the mesh based on the pose of the body, but does not take into account the deformations that define the identity of a person. To this end, we employ a global linear deformation model in which vertices $\boldsymbol{v}^{rest}_{i}$ are expressed as linear combinations $\boldsymbol{s}$ of bases stacked as columns into matrix $\boldsymbol{B}$: $\boldsymbol{v}^{rest}_{i} = \boldsymbol{m}_i + \boldsymbol{B}_i\boldsymbol{s}$. Previous work such as [ACP03, ASK*05, HSS*09] uses statistical models derived from a set of registered scans of people. The framework we propose allows for such models to be used (they use the same linear system), but in our implementation we employed blendshapes exported from the popular human body modeling software MakeHuman [BRM08]. These blendshapes correspond to the sliders in the MakeHuman application, that is used by numerous artists and game developers to generate realistic character assets. As a result, our model is based on a set of non-orthogonal bases (blendshapes) that are linearly combined in order to obtain novel shapes. The advantage of using such a technique as opposed to statistical models is that the fitting weights $\boldsymbol{s}$ represent a certain comprehensible body characteristic along each dimension (e.g.: *fat/slim middle*, *more/less muscular* etc.). This enables applications such as body shape retargetting, where the body parameters computed with our system could be used by artists to model bodies in

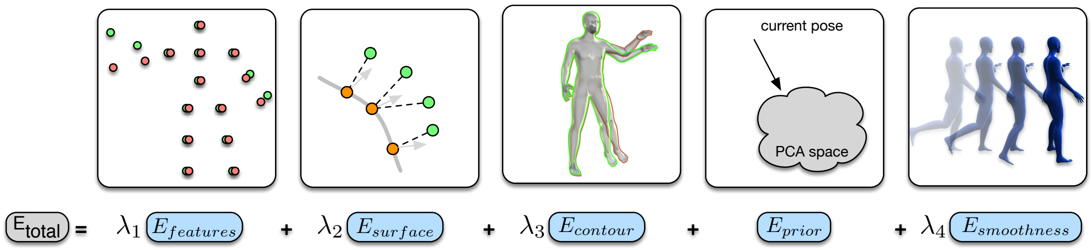

current pose
PCA space
$E_{total} = \lambda_1 E_{features} + \lambda_2 E_{surface} + \lambda_3 E_{contour} + E_{prior} + \lambda_4 E_{smoothness}$

Figure 3.3 – Visualization of the registration energies used in our optimization.

semantically equivalent spaces (e.g., use the human body blendshape weights to generate a semantically equivalent cartoonish character, animal or monster) (see Subsection 3.2.7 for more details).

We formulate the tracking and modeling problem as a global energy minimization, which aims to estimate the pose of the skeleton $\boldsymbol{\beta}_i$at each frame i, as well as the global shape of the body, encoded by $\boldsymbol{s}$: $\arg\min_{\{\boldsymbol{\beta}_i\},\boldsymbol{s}} E_{total}$. Figure 3.3 shows the different energies that we propose to use in our framework. In the following subsections we will explain the formulation of each energy functional, as well as offer an intuition on its contribution to solving the global problem.

3.2.2 Feature Constraints

In order to start with a good initial alignment and anchor the tracking, we use a soft energy that keeps the global translations of each joint close to the tracked sparse set of body landmarks. Several off-the-shelf solutions are available for tracking body landmarks over depth sequences. In our experiments, we have used the Primesense NiTE body tracker (currently not available due to the acquisition of the company). Alternatives are represented by the People Tracking module in PCL, as well as the Microsoft Kinect SDK[1]. All these trackers process, as input, a sequence of depth maps as those provided by a consumer depth camera, and output, at each frame, a set of tracked points representing the skeletal joints of the human body appearing in the sequence. As such, any of these methods could be used within our framework for the goal of tracking 3D body landmarks.

The energy term modeling the alignment between each template body joint and the respective estimated body landmark via tracking is formulated as follows:

$$E_{features} = \sum_j \left|\left| \boldsymbol{t}_j^{pose} - \boldsymbol{\chi}_j \right|\right|_2^2 \tag{3.1}$$

---

[1] https://www.microsoft.com/en-us/kinectforwindows/

where $\boldsymbol{t}_j^{pose}$ are the world-space translations of each joint, and $\boldsymbol{\chi}_j$ are the corresponding tracked 3D features.

3.2.3 Point-to-plane Constraints

The sampled scene surface obtained from the scanner is registered against the current estimate of the template body model. In order to align those two surfaces, the point-to-plane error metric is used:

$$E_{surface} = \sum_i ||\boldsymbol{n}_i^T(\boldsymbol{x}_i - \boldsymbol{v}_i)||_2^2 \tag{3.2}$$

where scan point $\boldsymbol{x}_i$ with its normal $\boldsymbol{n}_i$ is in correspondence with the template vertex $\boldsymbol{v}_i$. Mesh vertices $\boldsymbol{v}_i$ are expressed as functions of the skeleton pose and linear blend skinning, as explained in the previous subsection.

The PCL library offers multiple techniques for pre-processing the input data, obtaining and filtering pairs of corresponding points between the mesh and the depth map. The depth maps are represented as organized grids of 3D points, offering the possibility of using fast techniques such as integral images [HRD*12] to compute the normals of the depth maps efficiently. Furthermore, the input point cloud comes from a sensor that can be approximated via the pinhole camera model, the correspondences can be estimated in linear time by projecting the template vertices onto the depth map. Filtering is performed by discarding correspondences between points with incompatible distances and orientations [RL01]. The correspondences are computed at each outer iteration of the tracking optimization algorithm.

3.2.4 Contour Constraints

Due to the fact that the normals from depth data are noisy at the boundaries, they do not constrain the movement of the template body enough. To overcome this issue, an energy functional that minimizes the point-to-plane distances between the silhouette of the depth map and that of the template model is proposed.

From our experiments, we concluded that computing the silhouette by means of the following approach yields good enough quality for the purpose of our application (see the graphical example in figure 3.4, left). For the depth map, a pixel is considered to be on the boundary if it has less than 7 neighbors in its 3x3 neighborhood with a small depth difference (we used 3 cm in our experiments). For the template mesh, the boundary vertices are detected by rendering the current pose of the mesh in a framebuffer and extracting them

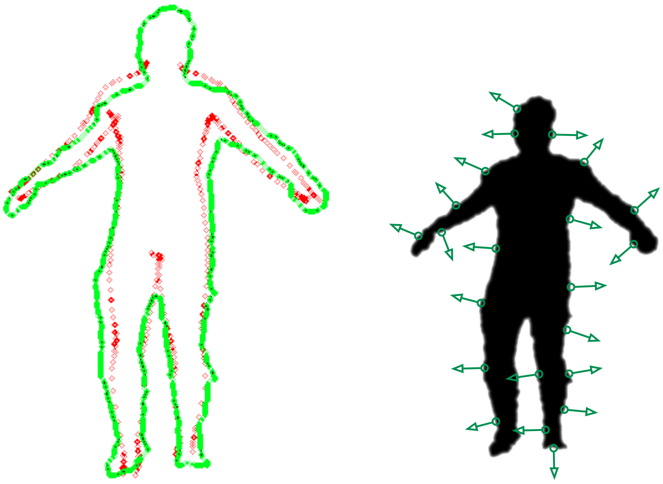


Figure 3.4 – Left: computed contour from depth and mesh: the green points represent the depth contour, the red points are the mesh contour, respectively. Right: associated normals computed on the depth map contour.

using morphological operators.

The energy functional for $E_{contour}$ is similar to the one in equation 3.2, with the differences that the correspondences are computed on the contour subsets as explained above, and the normals $\boldsymbol{n}_i$ (shown in green on the example contour image $\bar{\boldsymbol{I}}_{contour}$ in figure 3.4, right) are computed using the blurred contour image gradients as follows, with $\bar{\boldsymbol{I}}_{contour} = \boldsymbol{G}_{(\boldsymbol{\mu},\boldsymbol{\sigma})} \circ \boldsymbol{I}_{contour}$, and Gaussian kernel $\boldsymbol{G}_{(\boldsymbol{\mu},\boldsymbol{\sigma})}$:

$$\boldsymbol{n} = \frac{(\nabla_x \bar{\boldsymbol{I}}_{contour}, \nabla_y \bar{\boldsymbol{I}}_{contour}, 0)^T}{\left|\left|(\nabla_x \bar{\boldsymbol{I}}_{contour}, \nabla_y \bar{\boldsymbol{I}}_{contour}, 0)\right|\right|} \tag{3.3}$$

Furthermore, the correspondences are filtered by the angle between the projection of the template normals to the image plane and the depth pixel normals computed as above. Performing the normal computations and rejection step in 2D ensures more robustness to noisy input data.

3.2.5 Prior Energy

Principal Component Analysis (PCA) is a dimensionality reduction technique that has been employed numerous times in the tasks of modeling [BV99, ASK*05] and tracking [DOKA13]. In particular, Douvantzis et al. [DOKA13] use PCA to decrease the number of variables needed to describe the pose of a human hand. On the one hand, by doing so, the optimization problem

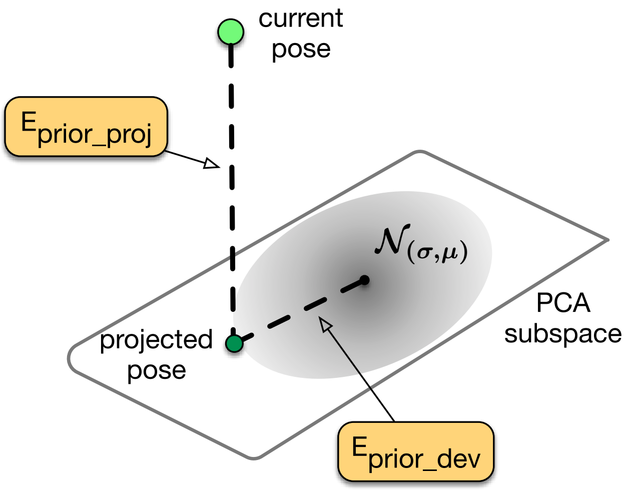


Figure 3.5 – Geometrical interpretation of the proposed PCA energy terms E_{prior_proj} and E_{prior_dev}.

becomes easier due to less variables that lie within trained statistical boundaries. On the other hand, solutions can only be picked from within the learnt subspace, limiting the tracking algorithm to be able to follow only poses similar to those in the training set. Furthermore, human bodies can undergo more complex pose changes as compared to hands, and it is considered very difficult to generate a comprehensive training set that contains all possible human poses. As such, in order to allow for novel poses to be tracked while still penalizing unlikely poses, our approach uses the PCA subspace as a regularizer instead of an optimization space.

To begin with, we train the PCA model by using multiple long sequences tracked using the NiTE feature tracker. This tracker is imprecise, but enough to enable the generation of a large collection of plausible human poses. The covariance matrix of the de-meaned data matrix $\boldsymbol{D}$ obtained by concatenating rows of joint angles $\boldsymbol{\beta}_j$ for each frame j in the training set is expressed using eigenvalue decomposition as $\boldsymbol{C} = (\boldsymbol{D} - \boldsymbol{1}\boldsymbol{\mu})^T(\boldsymbol{D} - \boldsymbol{1}\boldsymbol{\mu}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^{-1}$. $\boldsymbol{U}$ is the matrix formed of stacked columns of eigenvectors, $\boldsymbol{\Sigma}$ the eigenvalues in a diagonal matrix, $\boldsymbol{\mu}$ the mean of $\boldsymbol{\beta}$ over the training set. The eigenvectors are sorted in descending order by their corresponding eigenvalues and the first p modes are selected to form the PCA projection matrix $\boldsymbol{M}$. The number of modes p is chosen such that $\boldsymbol{M}$ forms a basis that explains a consistent portion of the training space (usually around 90%).

In order to keep the estimated skeleton pose $\boldsymbol{\beta}$ in the feasible space of poses, we introduce an error term that measures the distance between $\boldsymbol{\beta}$ and the back-projection of its projection to the PCA space:

$$E_{prior\_proj} = \left|\left|(\boldsymbol{\beta} - \boldsymbol{\mu}) - \boldsymbol{M}\boldsymbol{M}^T(\boldsymbol{\beta} - \boldsymbol{\mu})\right|\right|_2^2 \tag{3.4}$$

The previous energy tries to push the current estimate of the angles close to their projection in the PCA space. While this gives a soft guarantee that the current estimate of $\boldsymbol{\beta}$ can be expressed as the linear basis learnt in the PCA model, it does not regularize the values to the variance seen in the training set. To this end, we introduce an additional energy that penalizes the distance of the projections of $\boldsymbol{\beta}$ to the mean of the PCA space. The projection along each subspace dimension is weighted by the inverse of the standard deviation of the corresponding PCA basis, which is the square root of the elements in the diagonal eigenvalue matrix $\boldsymbol{\Sigma}$:

$$E_{prior\_dev} = \left|\left|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{M}^T(\boldsymbol{\beta} - \boldsymbol{\mu})\right|\right|_2^2 \tag{3.5}$$

Adding those two terms together, the prior energy functional with training becomes:

$$E_{prior} = \lambda_5 E_{prior\_proj} + \lambda_6 E_{prior\_dev} \tag{3.6}$$

A geometrical interpretation of the two terms E_{prior_proj} and E_{prior_dev} is given in figure 3.5.

In the case a training set is not available for computing the PCA model, an alternative prior energy $\tilde{E}_{prior}$ is used to keep the skeleton close to its neutral A/T-pose:

$$\tilde{E}_{prior} = \left|\left|\boldsymbol{\beta}\right|\right|_2^2 \tag{3.7}$$

However, it is important to point out that this method is less desirable than the data-driven technique described above as it allows for implausible human poses.

3.2.6 Smoothness Energy

Tracking each frame independently leads to jitter due to high frequency differences between the values of $\boldsymbol{\beta}$ from one frame to another. To overcome this, we introduce a soft contraint that penalizes large jumps of $\boldsymbol{\beta}$ between consecutive frames:

$$E_{smoothness} = \left|\left|\boldsymbol{\beta}_{(t)} - \boldsymbol{\beta}_{(t-1)}\right|\right|_2^2 \tag{3.8}$$

This is a first order smoothness term, enforcing that the angular velocity of the limbs is zero. More complex smoothness terms could be used in order to regularize the problem using acceleration or even higher order derivatives of the pose vector. We deemed the energy in equation 3.8 to be sufficient for the purpose of this system, as we do not expect excessively fast motions for the scenario of dynamic body scanning.

The smoothness energy term could be omitted, and the sequence could be smoothed as a post-processing stage, via temporal Laplacian filtering, for example. However, the smoothness energy included into the optimization acts as a soft prior term, adjusting the angular velocities of the limbs, and attracting the variables to the solution of the previous frame.

3.2.7 Tracking and Modeling

The optimization is split into two stages: *tracking*, aimed at estimating the pose $\boldsymbol{\beta}_i$ of the skeleton for each frame i, and *modeling*, aimed at estimating the body shape parameters $\boldsymbol{s}$ over the whole sequence. These two stages are briefly outlined in the following.

Tracking The tracking is performed sequentially for each frame, by keeping $\boldsymbol{s}$ fixed and finding values of $\boldsymbol{\beta}$ for which E_{total} is minimized. The Levenberg-Marquardt algorithm is used to optimize for the tracking, in which the linear system to be solved is computed analytically. The adaptive damping technique present in this algorithm is needed as the problem is unstable because of the chain of variables that influence all the nodes below in the tree. If the smoothness term $E_{smoothness}$ is removed from the total energy, then the tracking can be solved for each frame independently, allowing for heavy parallelization of this stage of the pipeline.

Modeling As mentioned before, the modeling stage refers to finding the optimal blend-shape weights $\boldsymbol{s}$ such that the mesh vertices $\boldsymbol{v}_i^{rest} = \boldsymbol{m}_i + \boldsymbol{B}_i \boldsymbol{s}$ minimize the modeling error over the sequence of frames tracked so far. Note that the modeling error does not contain some of the terms in E_{total}, as those were pertaining to regularizing the solutions for $\boldsymbol{\beta}$:

$$E_{modeling} = \gamma_1 E_{surface} + \gamma_2 E_{contour} + \gamma_3 E_{bs\_reg} \tag{3.9}$$

The tracking and modeling paradigm we proposed can be used for both online and offline modeling. Similar to [BWP13], one option is to use a temporal weighting scheme for the accumulation of the per frame constraints, giving more weight to more recent frames. In such a scheme, the modeling is done after each tracked frame, continuously updating the body model during the tracking. This is expressed mathematically as solving the linear system at time t, $\boldsymbol{M}_{lhs}^t \Delta \boldsymbol{s}^t = \boldsymbol{M}_{rhs}^t$. The update is done as follows, with:

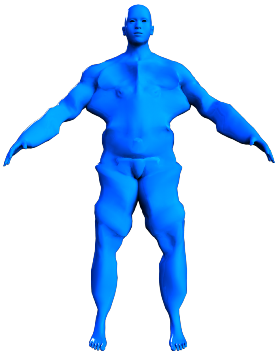


Figure 3.6 – Unrealistic body model obtained due to not regularizing the blendshape weights $\boldsymbol{s}$.

$$\boldsymbol{J}^t = \frac{\partial E^t_{modeling}}{\partial \boldsymbol{s}^t} \tag{3.10}$$

$$\boldsymbol{b}^t = E^t_{modeling} \tag{3.11}$$

the left and right-hand-side of the linearized constraints for frame t, respectively; w^t_γ is the temporal weight determined by the parameter $\gamma < 1$, which quantifies the influence of recent frames in the detriment of old frames:

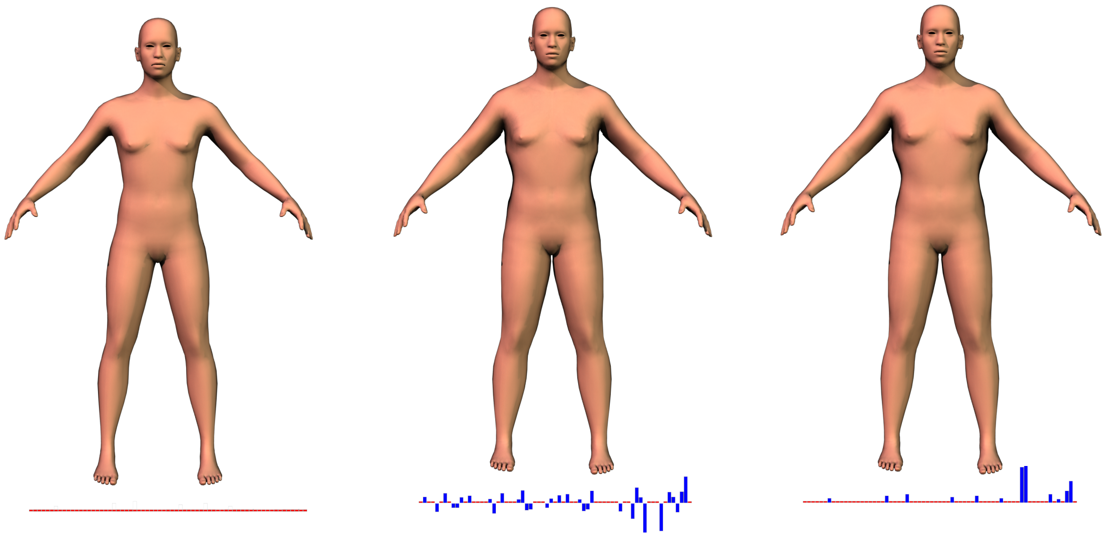
$$w^t_\gamma = \gamma w^{t-1}_\gamma + 1 \tag{3.12}$$

$$\boldsymbol{M}^t_{lhs} = \gamma \frac{w^{t-1}_\gamma}{w^t_\gamma} \boldsymbol{M}^{t-1}_{lhs} + \frac{1}{w^t_\gamma} (\boldsymbol{J}^t)^T \boldsymbol{J}^t \tag{3.13}$$

$$\boldsymbol{M}^t_{rhs} = \gamma \frac{w^{t-1}_\gamma}{w^t_\gamma} \boldsymbol{M}^{t-1}_{rhs} + \frac{1}{w^t_\gamma} (\boldsymbol{J}^t)^T \boldsymbol{b}^t \tag{3.14}$$

The second option is to accumulate all the frames of the sequence with equal weights ($\gamma = 1$ in the equations above), and solve for the modeling only at the end of the tracked input sequence. The main disadvantage of this scheme is that multiple tracking passes of the sequence are required, but the results will be more consistent with the whole dataset.

Allowing for any values of $\boldsymbol{s}$ can lead to unrealistic models such as the one depicted in Figure 3.6. Regularization is needed to keep the optimization from converging to such unwanted



(a) Template model
avg. tracking error: 0.39

(b) L2 fitting
avg. tracking error: 0.32

(c) L1 fitting
avg. tracking error: 0.34

Figure 3.7 – Visualization of the effect of the different norms on the blendshape weights for the body modeling of subject S4. Displayed are the resulting body meshes, along with a graphical representation of the blendshape weights and the average tracking error across the whole depth sequence obtained by using the respective body model estimation. (a) shows the scaled template model. (b) shows the modeling result with the classical L2 norm energy on the blendshape weights. (c) showcases the body model obtained by regularizing the blendshape weights with an L1 norm. The tracking error becomes slightly higher, but notice the smaller number of activated blendshapes with high values, compared to the L2 fitting, where a lot of blendshapes are activated with small values.

solutions. To this end, we suggest two options:

$$E_{bs\_reg\_L2} = ||\boldsymbol{s}||_2^2 \tag{3.15}$$

$$E_{bs\_reg\_L1} = ||\boldsymbol{s}||_1 \tag{3.16}$$

The effect of using either one of these regularization energies is explained in Figure 3.7. By employing $E_{bs\_reg\_L2}$, the linear system can be solved with the Gauss-Newton solver, with no damping necessary. Using the L1 norm present in $E_{bs\_reg\_L1}$ necessitates a different solver. In our implementation we use the Gauss-Seidel method with successive over-relaxation adapted to the L1 norm regularization [Fu98], along with iterative reprojections to keep the blendshape weights in the $[0,1]$ range. Another option, albeit usually considered slower, would have been Gauss-Newton with iterative reweighting [CY08].

In some situations the tracking does fail, and using those wrong constraints for the modeling phase might lead to erroneous results. In order to avoid such situations, we skip the frames for which the global tracking error E_{total} is higher than a certain threshold and the percentage of the overlap between the depth map and the template model surface is below a certain value. (in our experiments we chose 45%).

3.3 Implementation

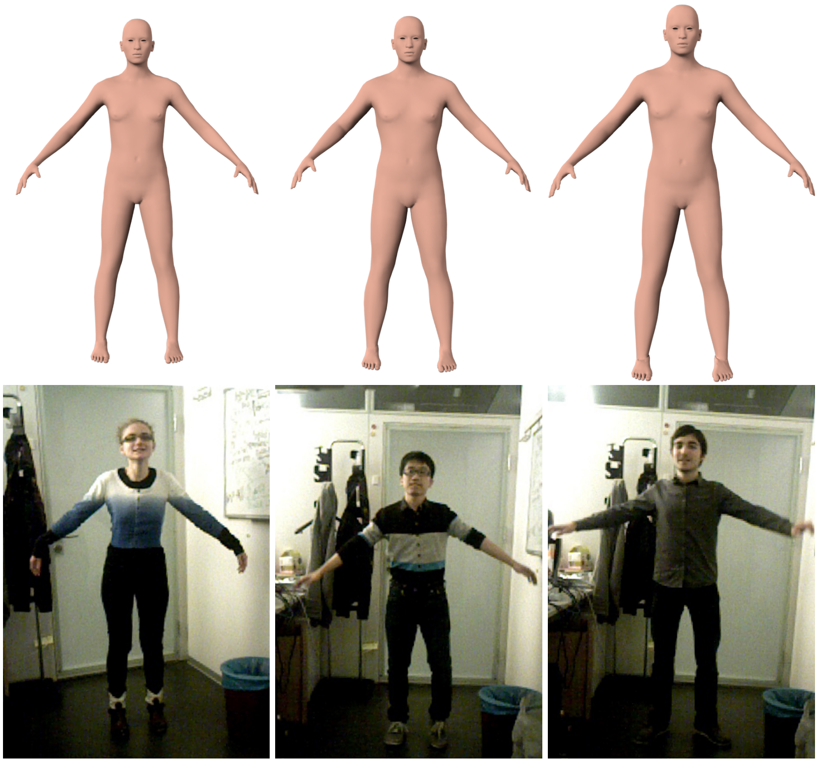
The skeleton is scaled once at the beginning of the pipeline by taking the median distances between the NiTe features at each frame over the whole sequence as being the limb lengths. From our experiments, this proved to be sufficient, and no further limb size adaptation was necessary during the tracking refinement and modeling iterations. Such a solution would not be possible in the case of online modeling and a different heuristic for the limb size adaptation should be employed.

The current implementation of the framework uses PCL exclusively, by relying on multiple components for the pre-processing, correspondence estimation and visualization stages. In particular, for efficient Nearest Neighbor search in the point cloud 3D domain we rely on the FLANN library included in PCL. Normals on the depth maps and on the point clouds are estimated, respectively, with the multithread method *pcl::NormalEstimationOMP* and with the integral images-based method *pcl::IntegralImageNormalEstimation*. All the solvers are implemented using the Eigen C++ linear algebra library, with no other external dependencies. For parallelization, we used the OpenMP library.

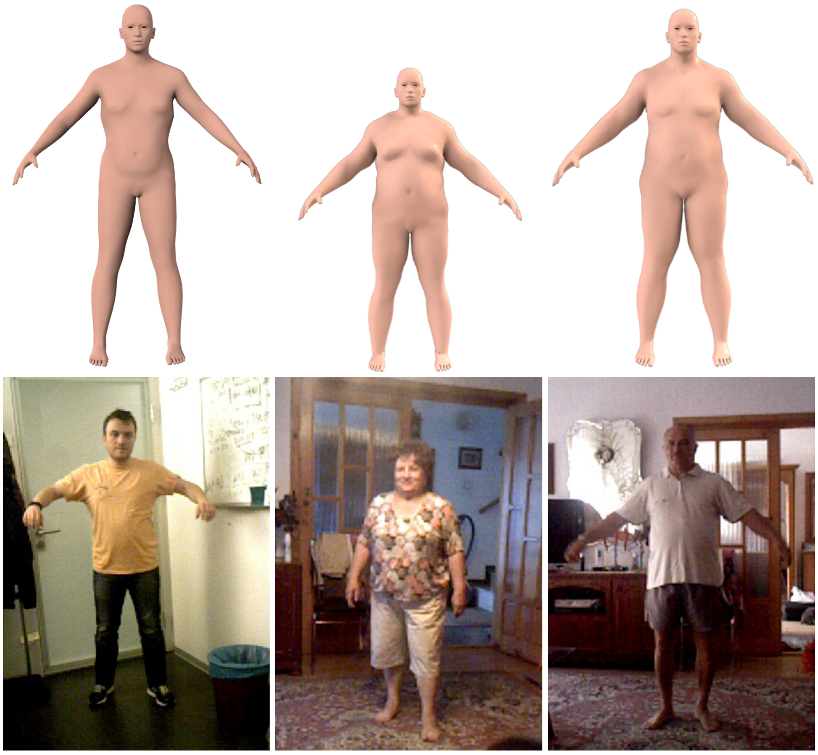
We have performed all of our experiments on data collected from an Asus Xtion PRO LIVE sensor, which consists of synchronized depth and color images at a resolution of 640x480 each, delivered at 30Hz. For the performed experiments, the blendshape model used for fitting contained 18 meshes selected from MakeHuman, representing macro shape variations of the body. The template mesh has 13380 vertices and 26756 triangles. The skeleton structure used is the *second_life* rig provided in the MakeHuman application, which has been manually mapped to the NiTE tracking features. The pose prior has been trained from a database of about 2300 frames with 18 3D feature locations per frame. The compressed PCA model retains 91% of the variability of the motion database by using 12 modes.

3.4 Experimental results

In this Section, we provide some experimental results and applications of our tracking and body shape estimation framework. To this goal, we have acquired multiple sequences of



(a) S1 (b) S2 (c) S3



(d) S4 (e) S5 (f) S6

Figure 3.8 – Modeling results showing the six experimental subjects ($S1 \cdots S6$) together with their modeled 3D body shape.

around 500-600 frames for each person of a group of four males and two female subjects (subjects $S1 \cdots S6$). Figure 3.8 shows one frame relatively from each acquired subject, together with the corresponding estimated body model. As it can be seen, the setting is that of a typical indoor environment, which includes cluttered background. As witnessed by the figure, the estimated body models are different to fit the specific body traits of each subject. In addition, we show, in Figure 3.9, multiple examples of tracked and estimated body models for different sequences, along with the corresponding RGB frames. As it can be seen, the method can track the 3D body also in complex poses, and also when the person's back is facing the camera. For a more clear view of the incremental tracking and modeling process, Figure 3.10 shows the evolution of the template silhouette with respect to the depth map contour as the optimization converges.

To evaluate quantitatively the modeling precision of our framework, we computed the standard deviations measured on a few key locations of the body model estimated along each of the 4 sequences associated to each of the 4 subjects for which such number of sequences was available (i.e., subjects $S1 \cdots S4$). The results are reported in Table 3.1, along with a visualization of the location of the body features (on the left). The reported results range between a minimum of 0.36 cm and a maximum of 2.35 cm, with an average over all locations and all subjects of 1.10 cm, which demonstrates an encouraging repeatability of the proposed modeling algorithm, leading us to believe that this framework has the potential to obtain relatively accurate results, even without the need of a complicated setup or scripted actor movement.

We wish to point out here that our framework does not explicitly take into account the presence of clothes. Indeed, precise body pose measurements should be taken without clothes or wearing garments that are tight to the body. Indeed, loose clothes might easily lead to errors in both the tracking and modeling stages of the pipeline, this resulting in decreasing the accuracy of body measurements. The same problem would occur if the subject has long/voluminous hair, or in presence of any accessory with a non negligible size, such as bags, hats, glasses, belts, etc.

As mentioned before, the L1 semantic body modeling we propose opens up multiple avenues for applications that could have not been possible with L2 statistical models. A simple such example is depicted in Figure 3.11, where the word clouds corresponding to the blendshape weights of the body fitted in Figure 3.7 have been created. Note the compactness of the semantic representation yielded by the use of the L1 relaxation with respect to the one yielded by the L2 model.

Moreover, by having the body mesh modeled and tracked throughout the sequence, and

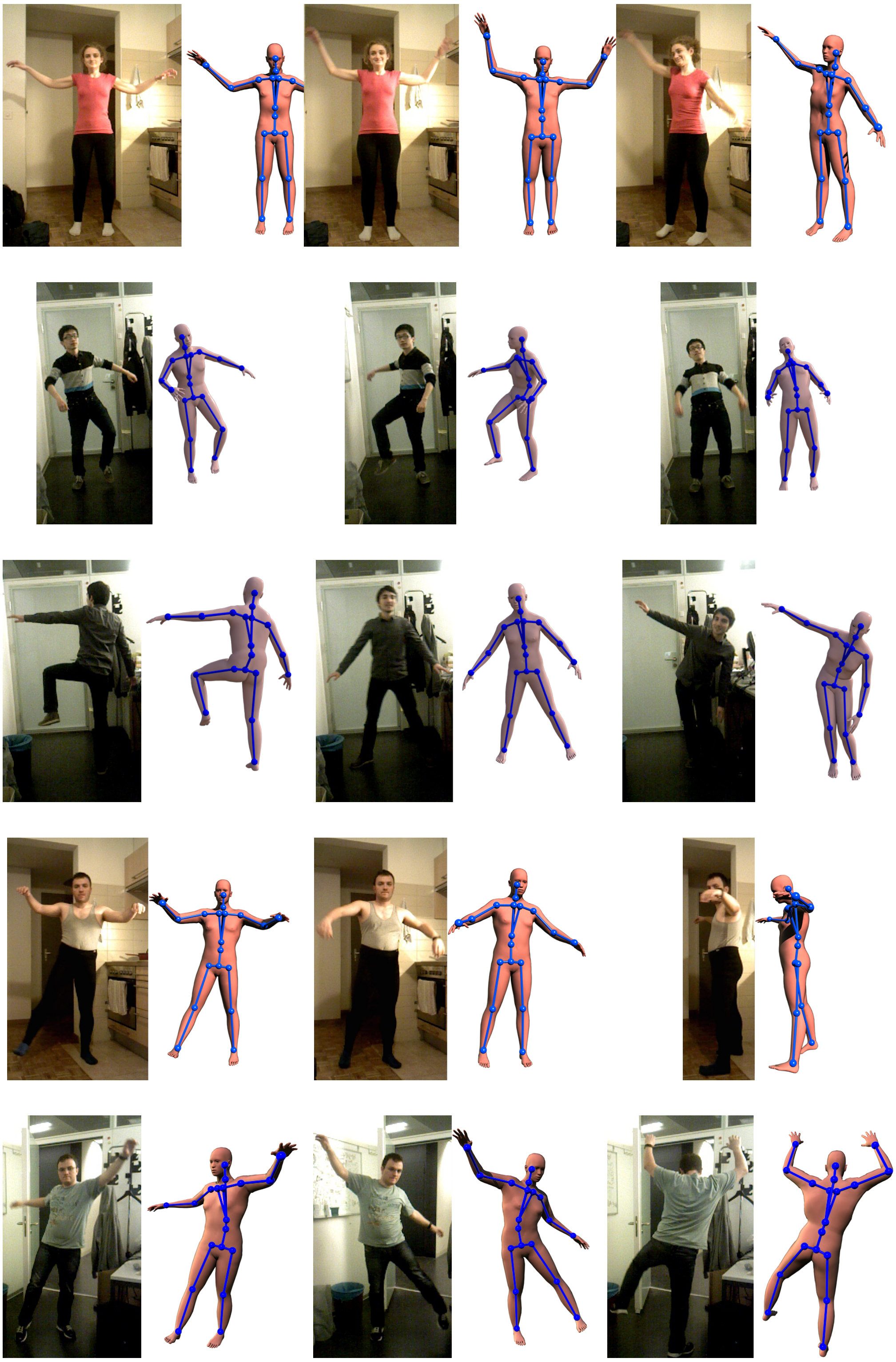


Figure 3.9 – The proposed framework is able to track bodies and estimate their parametric body model from a depth sequence. In each row of the figure, we report three examples of a sequence corresponding to one of the evaluated subjects (S1, S2, S3, S4, and S4, respectively). For each example, we show the input RGB frames, the posed body and its corresponding skeleton.

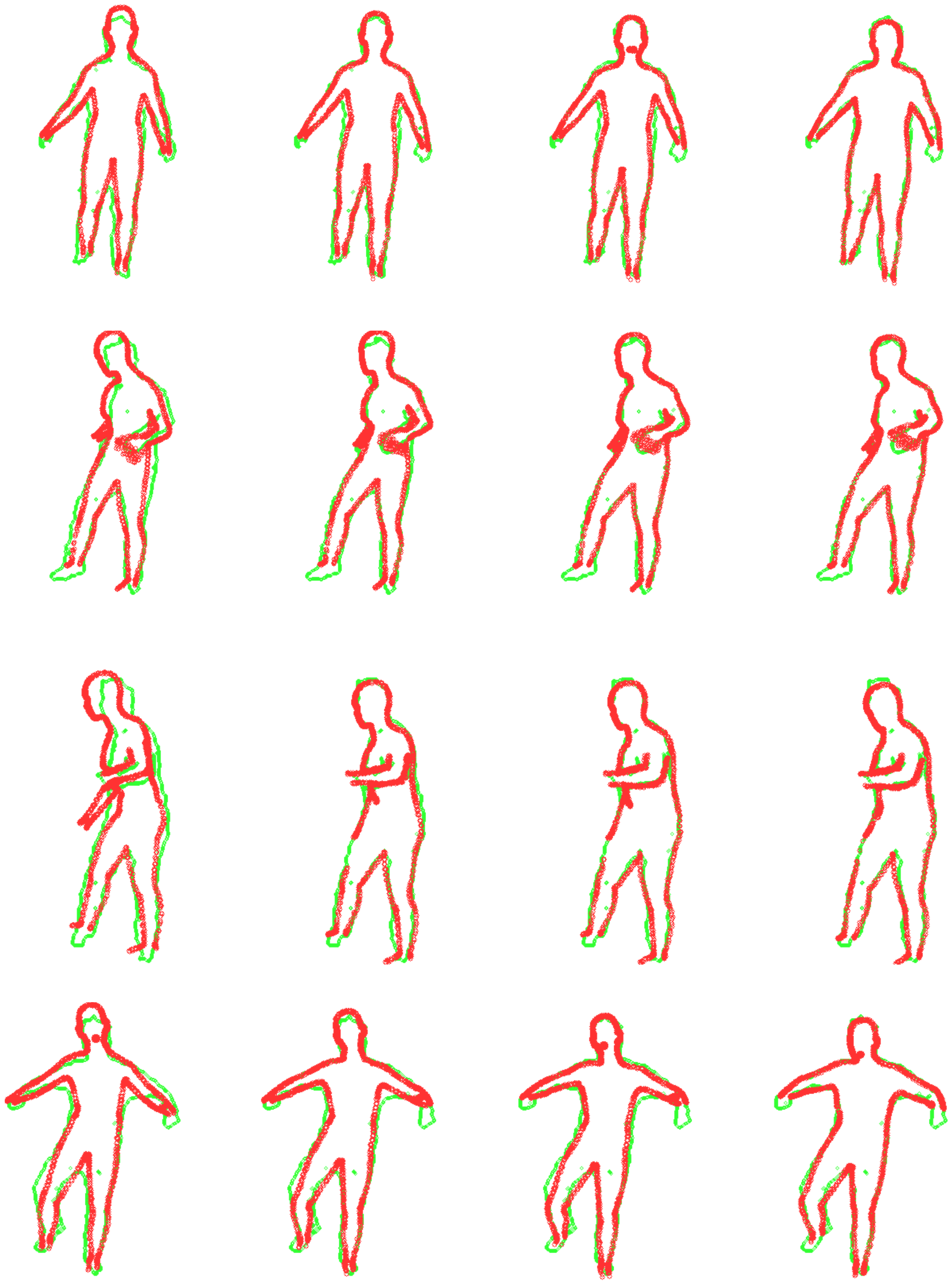
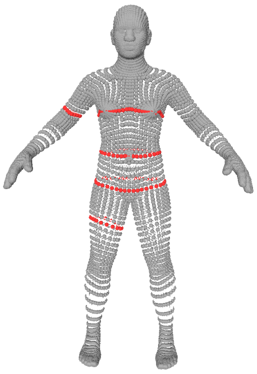


Figure 3.10 – Each row of images shows the progress of the tracking and modeling optimization for the contours of single frames of actor S4, starting from the initial estimate given by the 3D feature locations and the template mesh, up to convergence for both the pose and the body shape parameters. The template mesh silhouette is drawn in red, and the depth map contour is green.



	S_1		S_2		S_3		S_4	
	μ	σ	μ	σ	μ	σ	μ	σ
arm	25.97	0.35	28.95	0.42	29.4	0.36	32.55	1.05
chest	89.59	0.70	100.96	2.35	105.35	0.6	108.68	1.01
hips	92.27	1.24	99.37	1.01	105.21	0.59	106.65	0.67
leg	48.74	1.07	54.91	0.49	55.6	0.91	59.56	1.22
waist	76.70	1.53	83.2	1.92	91.25	0.84	94.67	1.12
avg		1.06		1.46		0.69		1.05
total avg σ	1.10							

Table 3.1 – Mean and average standard deviation of all the measurements for 4 test subjects with 4 recorded sequences each. All reported values are in cm. On the left: locations of the five measurements taken on the subjects

l-lowerarm-skinny
l-lowerarm-muscle-decr
r-upperleg-skinny
l-lowerleg-muscle-decr
r-leg-genu-valgus
l-lowerleg-skinny
l-leg-genu-varun
stomach-pregnant-incr
bulge-decr
buttocks-volume-decr
bulge-incr
universal-female-young-maxmuscle-minweight
r-upperarm-muscle-decr
l-upperarm-skinny
l-upperarm-muscle-decr
stomach-pregnant-decr
stomach-tone-incr
universal-female-young-averagemuscle-minweight
r-upperarm-shoulder-muscle-incr
stomach-navel-up
l-upperarm-fat
l-lowerarm-fat
torso-vshape-more
l-lowerleg-fat
r-upperarm-muscle-incr
r-lowerarm-muscle-incr
r-upperleg-muscle-decr
l-upperarm-muscle-incr
r-lowerleg-fat
r-leg-genu-varun
r-lowerleg-skinny
stomach-navel-down
r-upperarm-fat
r-lowerarm-skinny
l-upperleg-muscle-decr
torso-vshape-less
l-upperleg-muscle-incr
r-lowerarm-fat
l-lowerleg-muscle-incr
pelvis-tone-decr
universal-female-young-maxmuscle-maxweight
l-lowerarm-muscle-incr
buttocks-volume-incr
r-upperarm-shoulder-muscle-decr
r-upperarm-skinny
l-upperleg-fat
pelvis-tone-incr
universal-female-young-minmuscle-minweight
r-upperleg-muscle-incr
l-upperleg-skinny
l-leg-genu-valgus
universal-female-young-averagemuscle-minweight
universal-female-young-maxmuscle-maxweight
universal-female-young-minmuscle-minweight
r-upperleg-fat
pelvis-tone-incr
universal-female-young-minmuscle-averageweight
torso-muscle-dorsi-incr
l-upperleg-fat
torso-vshape-less
torso-muscle-pectoral-decr
l-leg-genu-varun
universal-female-young-minmuscle-maxweight
l-lowerarm-fat
r-upperarm-fat
stomach-pregnant-decr

(a) (b)

Figure 3.11 – Example application of the proposed semantic body modeling. (a), (b) show word clouds created using the activated blendshape names and their weights for the L2 blendshapes regularization, and the L1, respectively.

being the associated RGB frame available as well in correspondence with each depth frame (i.e., RGB-D data), our framework allows to build a complete texture of the mesh (see Figure 3.12). A simple technique is employed that projects each RGB frame into the UV-space of the mesh, accumulating color contributions weighted by the foreshortening angle (angle between the normal at the mesh surface and the viewing direction of the camera). Note the presence of artifacts such as blurring and black regions. These are due to inaccuracies in the tracking and the fact there are regions that have not been captured during the sequence.

3.4.1 Effect of each tracking energy

Finally, in Figure 3.13 we show some results relatively to the influence of the most relevant energy terms employed in the proposed cost function, obtained by deactivating them one at a time and highlighting the most remarkable qualitative differences. As witnessed by 3.13a, the **surface registration energy** encapsules the most important set of constraints of the proposed optimization. Its purpose is to align the underlying scene surface sampled by the depth sensor

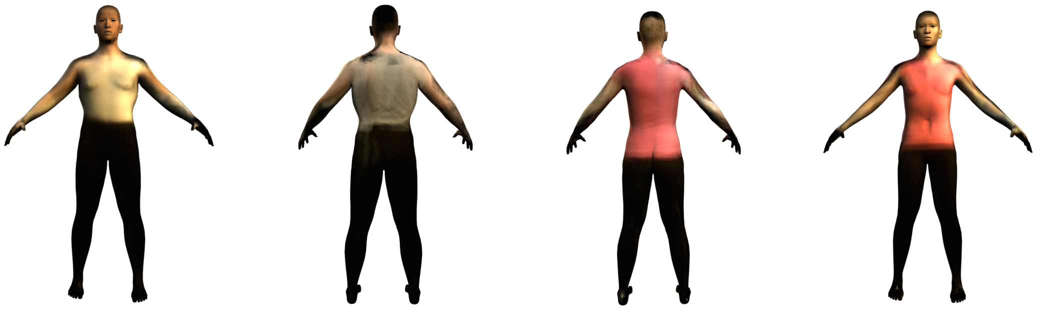
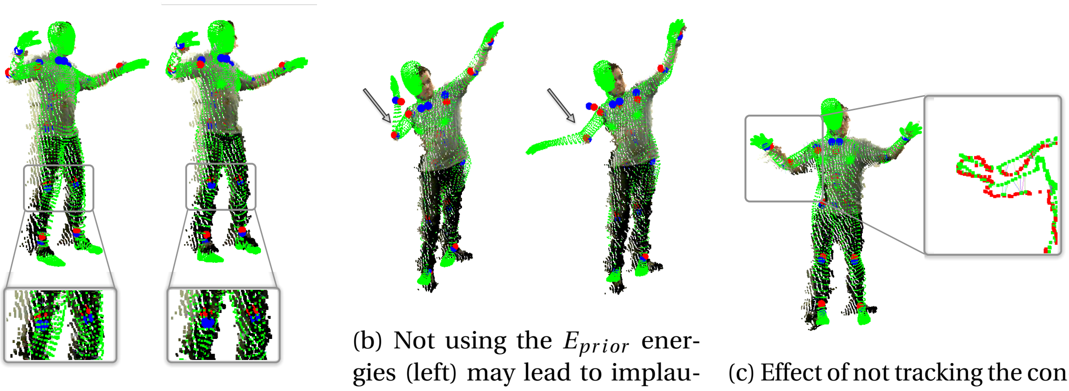


Figure 3.12 – Our framework allows for texturing the body meshes using a weighted average of the color contributions from each RGB frame.



(a) Without and with $E_{surface}$.

(b) Not using the E_{prior} energies (left) may lead to implausible poses.

(c) Effect of not tracking the contours.

Figure 3.13 – Influence of the proposed energy terms on the resulting tracking.

with the template mesh. The contour and feature constraints in addition to the tracking priors are not enough for precise tracking. Differently, the **contour energy** is useful because the depth map normals are rather flat at the silhouette of the objects in the scene, thus not constraining the tracking enough in those regions. As such, misalignments like the one in figure 3.13c can occur as the point to plane energy is minimized even if the hand alignment is off. The contour correspondences shown in the zoomed in part of the figure would have pulled the template body to its correct location. These effects can accumulate in time and lead to a complete loss of tracking.

Moreover, the **prior energy** is needed in order to avoid implausible poses to be outputted. In the situations when the input data is lacking information about certain regions of the body due to occlussions, the statistical pose priors we propose help keep the body in a reasonable pose. An example is shown in figure 3.13b, where the 3D features were wrong. Without priors the tracker moves the arm in an impossible pose, but employing the priors (right side) converged to a more plausible solution. Finally, although not shown in the Figure, the **feature energy** that keeps the joint position close to the NiTE 3D features detected for each frame is

Step	**Average number of iterations**	**Average time [ms]**
Normal estimation	1	24
Initialize IK with 3D Features	1	1
Point to plane correspondences	3.5	3
Contour correspondences	3.5	26
IK with all energies	3.5	27
Accumulate modeling constraints	1	3
Tracking per frame	-	279
Modeling per frame	-	3

Table 3.2 – Performance benchmarking results for each stage of the pipeline. The average number of iterations is per frame. The rest of the total per frame average time is spent with other book-keeping operations.

especially useful for initial alignment, after which its weight can be decreased to zero during the optimization. Without it, manual intervention would be mandatory to pose the body so that the iterative optimization tracking can have a warm start.

3.4.2 Performance Evaluation

We have performed our experiments on a laptop with an 8-core Intel Core i7-4940MX processor, with 32GB of RAM, and a GeForce GTX 880M graphics card, running Ubuntu 14.10. The code uses the CPU for all of the computation, with the exception of the framebuffer rendering for extracting the mesh silhouette, which is done using primitive OpenGL calls. Table 3.2 collects the timing information from our experiments. Furthermore, it is worth mentioning that the IK optimization using only the 3D features was tuned to use an average number of 24.1 Levenberg Marquardt iterations, and the IK optimization with all the energies performed with an average of 10.44 internal iterations.

3.5 Conluding Remarks

In this paper we have presented a modular framework for 3D body tracking and parametric modeling. Our framework can run efficiently, and thanks to the use of MakeHuman models together with L1 relaxation allows for new applications such as semantic body part tagging of the acquired subjects, as well as body model texturing. Moreover, our framework can be easily adapted to fitting posed bodies that have been scanned using technologies such as PCL's Kinect Fusion [NIH*11] implementation, KinFu, or multiview reconstruction using external tools such as 123DCatch [BF14].

The code is modularized in a logical structure that allows for further experimentation and extensions. We are planning on releasing the code as open-source, and integrate it in the *pcl::bodies* module of the PCL library. Indeed, we believe that our contribution can lead to a higher interest from developers and researchers to dive into more complex body tracking and modeling applications. Also, we believe that due to its customizability, it will enable future work which includes novel research, better benchmarking data and interesting new applications.

Retrospective

Because this work has been intended as a generic framework for body tracking, the different modules that build the optimization constraints can be replaced and tuned for various sensors and applications, as well as updated in order to reflective novel, more performant techniques. For example, one could use a more sophisticated approach for landmark detection such as ones based on different deep learning techniques [WHC*16, MSS*17].

It is worth mentioning that the parametric body model can be replaced with other techniques that could offer more robustness [ASK*05], or added value such as intrinsically modeling learnt secondary motion [PMRMB15]. Moreover, recent techniques have shown that one can reconstruct good quality models of general deformable scenes [NFS15, ZNI*14], and it would be interesting to see how these could be integrated with our body tracking and modeling framework.

In the next chapter we present our subsequent work that turns away from linear skinning models and investigates how volumetric anatomical templates can be used within a physics simulation framework in order to better model and animate human bodies. While we have not directly experimented with this, our newer work [KIL*16] can be adapted to RGB-D body tracking by following this framework. However, as opposed to linear skinning models, the complexity of the optimization will not make this combination suitable for realtime applications. Furthermore, we believe that the low quality of the data coming from commercial RGB-D sensors does not justify the need for more complex models. In order to benefit from the increased representational power of physics-based models, in the next chapter we only perform experiments using high-quality 3D scans.

4 Reconstructing Personalized Anatomical Models for Physics-based Body Animation

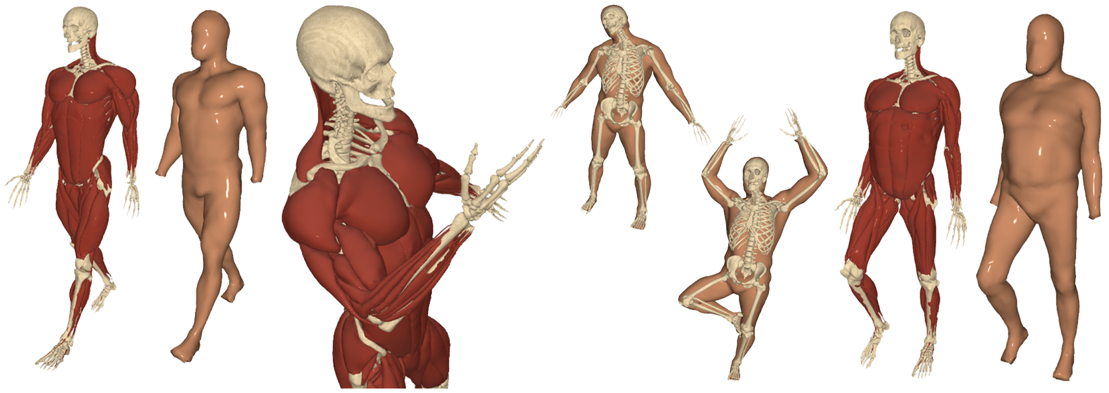


Figure 4.1 – We present a full-body reconstruction and animation system that can simulate physics-based volumetric effects such as self-collision and inertial effects. Our method uses a set of 3D surface scans to adapt an anatomically-inspired volumetric model to the user.

Note

This chapter corresponds to the following publication [KIL* 16]:

KADLECEK, P. (*), ICHIM, A.E. (*), LIU, T., KAVAN, L., AND KRIVANEK, J. (* joint first authors). Reconstructing Personalized Anatomical Models for Physics-based Body Animation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia), 2016*

The first authorship for this paper has been shared with Petr Kadlecek. The author's contributions are as follows:

- inverse body modeling formulation and optimization solution
- roughly half of the implementation effort.

Abstract

We present a method to create personalized anatomical models ready for physics-based animation, using only on a set of surface 3D 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 solve for 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 the Newtonian dynamics.

4.1 Introduction

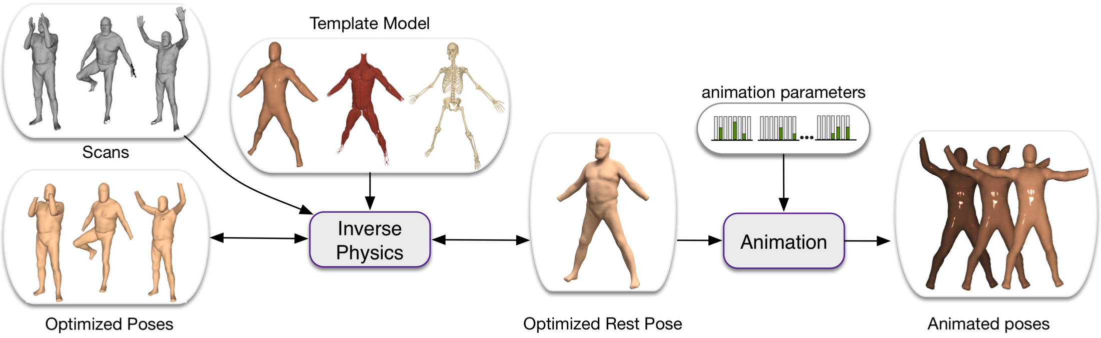


Figure 4.2 – Workflow of our method: We take as input a set of 3D scans of the same actor in different poses. Our method aims at reconstructing a complete volumetric, rigged, and physics-ready body model of the actor. This consists of extracting its exterior and interior shape, as well as skeleton bone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

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 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 paper we present an automatic method to create an anatomical, physics-based model of the body of a given human subject, e.g., an actor. We achieve this by capturing a set of 3D full-body 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 [ASK*05] or the more recent BlendSCAPE [HLRB12] (to name just a two) construct an articulated human body model from a set of input 3D scans. Similarly to artist-directed systems such as Pose Space Deformation [LCF00], these methods build a data-driven model which predicts skin deformations based on the skeletal pose (joint rotations). However, these methods focus exclusively on the skin, i.e., outer boundary of the human body. The skeleton is modeled as connected line segments, disregarding the volumetric nature of bones or even muscles. While surface-based data-driven methods are effective in 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 [PMRMB15], 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 (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. 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 be of course combined with other computer graphics techniques such as displacement mapping in order to model, e.g., prominent veins.

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 paper), 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 [SZK15]. While Computational Bodybuilding presented methods for the *forward* simulation of growth of bones, muscles, and adipose tissues, in this paper, we study the *inverse* problem, i.e., we solve an optimization problem to recover the growth 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 action-reaction internal forces (when one bone/muscle grows, it pushes the adjacent organs out of the way).

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. Most parts of our forward skinning model (Section 4.4) are derived from previous work, however, we had to devise a new elastic potential (which we call "symmetric as-rigid-as-possible" energy) in order to make the subsequent inverse modeling work (classical as-rigid-as-possible models did not work, as discussed 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. [ASK*05] 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. An advantage of our method over this is the fact that the same deformation model is used for all the people, while we construct person-specific internal components. This data-driven approach was extended and applied to sparse motion-capture animation in Loper et al. [LMB14], in order to obtain better quality motion reconstructions as compared to traditional skeleton-driven skinning approaches. Zuffi et al. [ZB15] 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, [PMRMB15] introduces a novel 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 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. [TMB14] 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 [DAA*07] 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.

The survey of [LGK*10] 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 physical-based and data-driven approaches are the ones of most interest for our work. Teran et al. [TBHF03, TSIF05, TSB*05] 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.

[SZK15] 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 [BML*14]. The key difference from our method is that [SZK15] requires the bone/muscle/fat growth parameters to be provided by the user.

Fan et al. [FLP14] propose a framework for simulating a dynamic volumetric muscoskeletal system using an Eulerian-on-Lagrangian discretization that can handle sliding elements in close contact, volume preservation and large deformations.

Anatomy Transfer [DLG*13] 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 register it to new surface-mesh humanoid models. 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, [ZHK15] adapts the bone structure of upper and lower limbs given an RGB-D sequence of moving limbs.

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. [SLST14] 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. [LT06] 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. [BBO*09]. They used a scattered data interpolation technique in strain-space to simulate novel deformations of objects composed of the observed materials. Similarly, [WWY*15] use off-the-shelf 3D sensors to track and model deformations of soft objects using physics-based probabilistic priors.

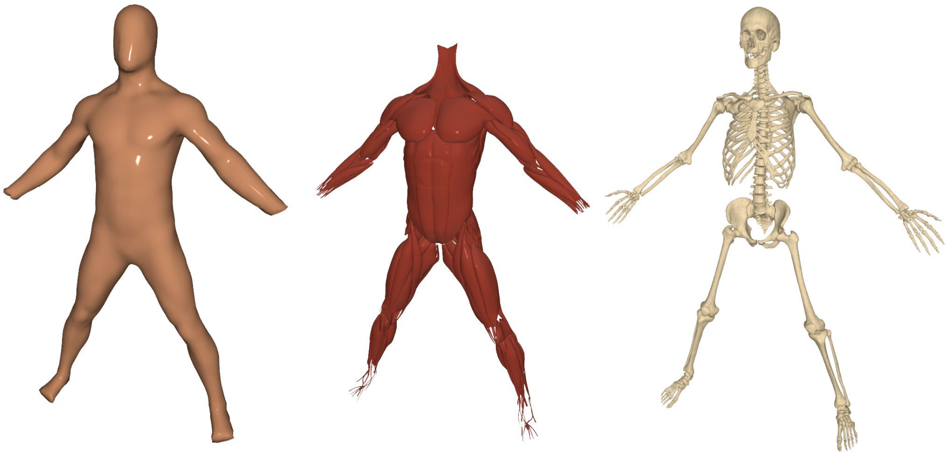


Figure 4.3 – Components of our anatomically-inspired volumetric template model.

[CZXZ14] 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.

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 Figure 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. [SZK15] by starting from the commercially available Zygote anatomical body model [Zyg16] with 111 muscles and 204 bones represented as meshes. The skin, as well as the muscles and bones are uniformly remeshed with the Instant Meshes algorithm [JTPSH15] and then the surfaces are tetrahedralized using the approach of Jacobson et al. [JKSH13].

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 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 4.6), the muscle fiber directions are required (see Figure 4.4). We compute the fiber directions in a similar way as Saito et al. [SZK15]. First, the tendon regions are selected manually and associated with Dirchlet boundary conditions. The non-tendon muscle boundaries are associated with Neumann boundary conditions. Next, we solve a Poisson equation for a

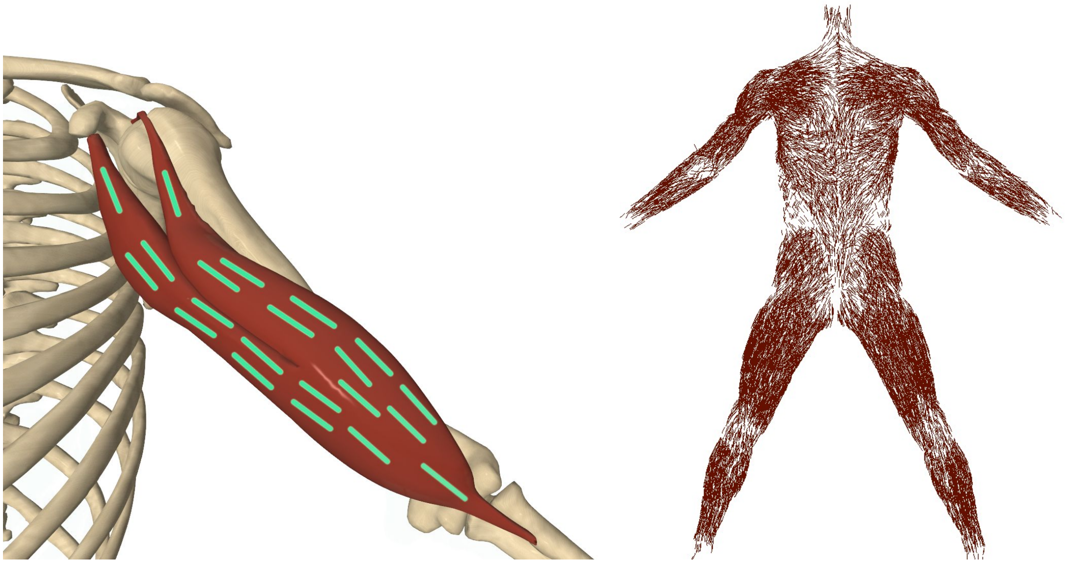


Figure 4.4 – Left: a close-up on the fibers on the right bicep muscle. Right: Visualization of the embedded muscle fibers in the template model.

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 "muscle envelope," [SZK15], 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 [WSA*02, WVdHV*05]. 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.

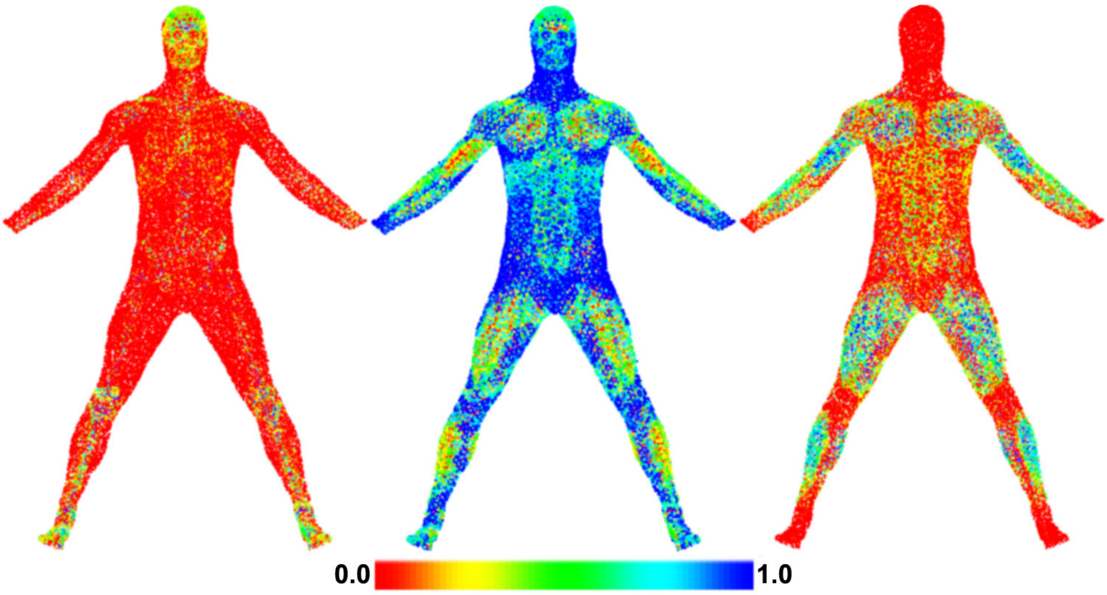


Figure 4.5 – The distribution of the material types inside the body. From left to right: bones, generic soft tissue, muscle.

4.4 Forward Skinning Model

Before diving into the *inverse* problem of body reconstruction, we will first describe our *forward* physics-based character model. Our model is built by extending recent works, in particular Saito et al. [SZK15] and Zhu et al. [ZHK15]. Saito et al. simulated growth only in the rest pose, skeletal rig was not included. Zhu et al. did create a skeletal rig, but only for the extremities: the arm and the leg. Also, the deformation model of Zhu et al. [ZHK15] was based on direct skinning models. In this paper, 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* [MZS*11]: 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:

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

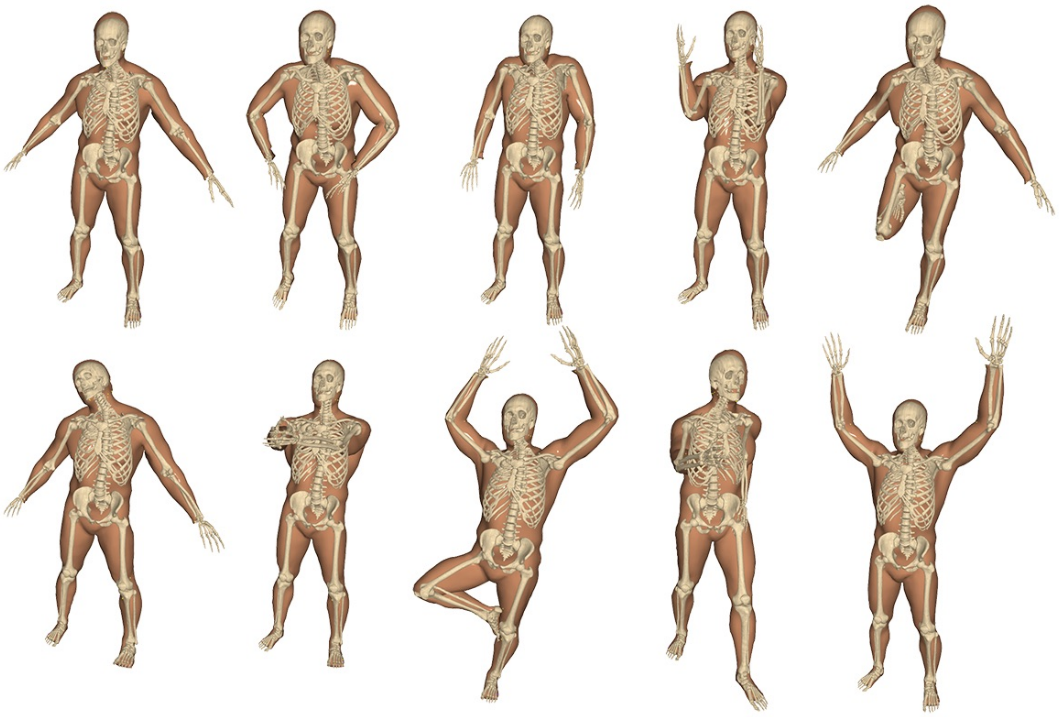


Figure 4.6 – Complex skeleton rig fitting on Faust dataset.

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 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. $E_{\mathrm{def}}(\mathbf{X}^{\mathrm{src}}, \mathbf{X})$ is an elastic potential function which measures the amount of deformation between configurations $\mathbf{X}^{\mathrm{src}}$ and $\mathbf{X}$ (both of which correspond to meshes with the same connectivity). $E_{\mathrm{grav}}(\mathbf{X})$ 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 fat man jumping. Finally, $E_{\mathrm{col}}(\mathbf{X})$ is energy potential penalizing collisions, i.e., self-intersections of the mesh, see Section 4.5.1 for more details on collision processing. The necessary condition for $\mathbf{X}$ being in quasi-static equilibrium is $\nabla_{\mathbf{X}} E_{\mathrm{skin}} = 0$, i.e., sum of forces is zero. More details on the above mentioned terms follow.

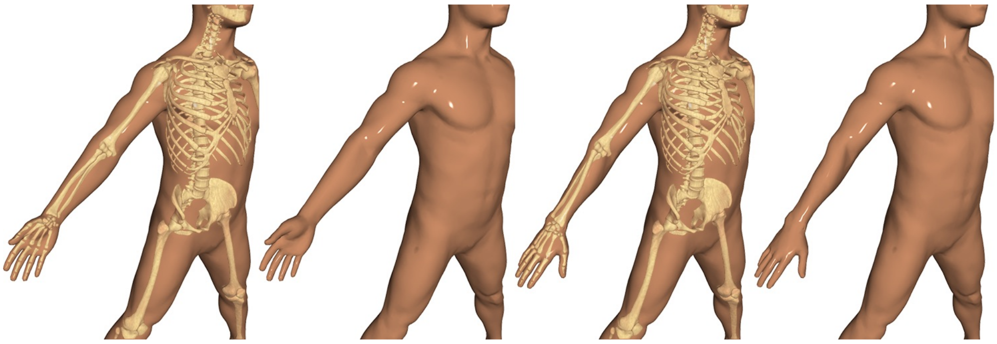


Figure 4.7 – Complex pronation-supination motion is handled well by our physics skinning.

BoneFlesh. The BoneFlesh term introduced above 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:

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

where $\mathbf{S}^{\mathrm{bone}}$ 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 [Si15]. 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).

Rig. Our kinematic skeleton model is modeled by function $\mathrm{Rig}(\boldsymbol{\theta},\boldsymbol{\pi})$, 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 function Rig 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 [LT08]) could be added to improve the accuracy of the kinematic modeling.

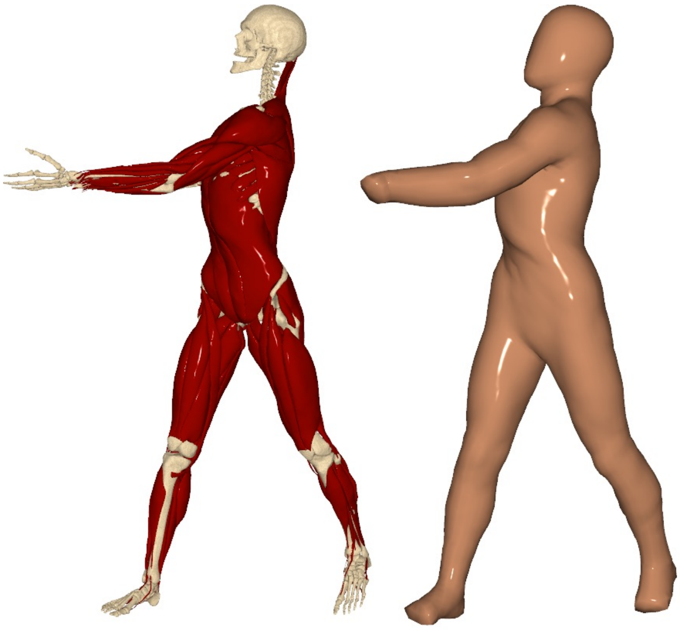


Figure 4.8 – Anatomically correct bones produce more realistic body shapes e.g. during upper trunk rotation, where the rib cage retains its shape.

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. [ZHK15]. Specifically, each bone is deformed using linear blend skinning with bounded biharmonic weights [JBPS11] with handles located in the center of each of the bone heads, see Figure 4.10. The handles of adjacent bones (i.e., forming a joint) are constrained to be transformed by the same matrix which contains only rotation, translation and uniform scale. This guarantees that the structure of the joint will be preserved. Formally, we can express this deformation using function $\mathrm{BoneGrow}(\boldsymbol{\pi})$ which depends only on the growth parameters $\boldsymbol{\pi}$ and produces the modified rest pose bone vertex samples.

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 [MLSS94], 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 $\mathrm{FK}(\boldsymbol{\theta},\boldsymbol{\pi})$, the entire rig function can be written as composition:

$$\mathrm{Rig}(\boldsymbol{\theta},\boldsymbol{\pi}) = \mathrm{FK}(\boldsymbol{\theta},\boldsymbol{\pi})\mathrm{BoneGrow}(\boldsymbol{\pi}), \tag{4.4}$$

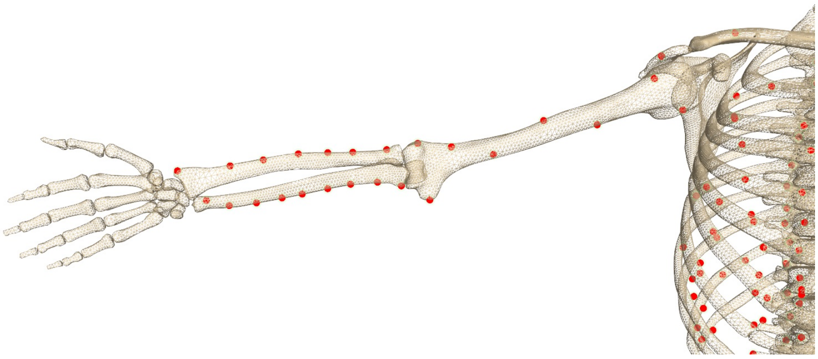


Figure 4.9 – Sample bone vertices corresponding to the selector matrix $\mathbf{S}^{\text{bone}}$.

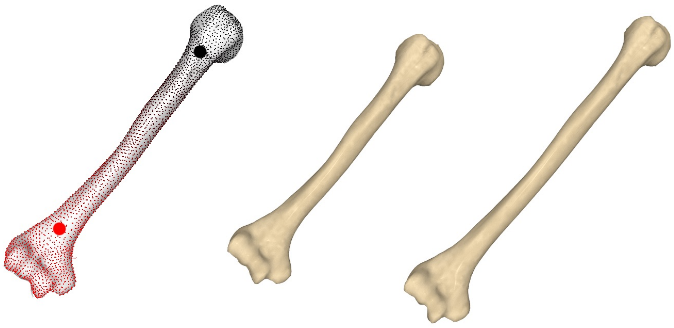


Figure 4.10 – Example of the humerus bone elongation preserving shape of bone heads using two deformation handles and precomputed bounded biharmonic weights.

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 BoneGrow.

Elastic Potential E_{def}. Elastic models of biological soft tissues have received considerable attention both in the biomechanics [WMG96, Fun13] as well as computer graphics communities [TSB*05, TSIF05, SNF05, LST09, SB12]. Neohookean hyper-elastic materials have been found to function well in recent work [BKS*12, STK*14, STC*13]. 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. [ITF04], inverted tetrahedra may be necessary to capture large deformations without resorting to remeshing. Increasing the mesh resolution (e.g., subdividing tetrahedra) 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 [SB12]). 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 [MZS*11], 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.

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

$$E_{\text{def}}(\mathbf{X}^{\text{src}}, \mathbf{X}) = \sum_i k_i \|\mathbf{D}_{S,i} - \mathbf{R}_i\mathbf{D}_{M,i}\|_F^2, \tag{4.5}$$

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 $\mathbf{X}^{\text{src}}$, $\mathbf{D}_{S,i}$ depends linearly on $\mathbf{X}$ 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-possibe" 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 $\frac{1}{6}|\det(\mathbf{D}_{M,i})|$ [SB12], 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.

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. [LST09], 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 4.8 for more details.

Muscle growth. Our symmetric as-rigid-as-possible (ARAP) elastic model can be extended to account for muscle growth [SZK15]. We accomplish this using the following energy:

$$E_{\text{muscle}}(\mathbf{X}^{\text{src}}, \mathbf{X}, \alpha) = \|\mathbf{D}_S - \mathbf{R}\mathbf{B}\mathbf{S}(\alpha)\mathbf{B}^{\mathsf{T}}\mathbf{D}_M\|_F^2, \tag{4.6}$$

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

$$\mathbf{S}(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{pmatrix}. \tag{4.7}$$

4.5 Inverse Body Modeling

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 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 standard non-rigid ICP procedure [RL01], explained in more detail in Section 4.5.2. We denote the resulting registered meshes as $\mathbf{T}_i$, where $i = 1 \ldots \text{numScans}$. 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 growth parameters $\boldsymbol{\pi}$ and joint angles $\boldsymbol{\theta}_i$, where i also indexes individual poses, $i = 1 \ldots \text{numScans}$. The growth parameters $\boldsymbol{\pi}$ are fixed for a given human being, but the joint angles $\boldsymbol{\theta}_i$ vary from pose to pose. We need to find the values of $\mathbf{X}^{\text{pers}}$, $\boldsymbol{\pi}$, and $\boldsymbol{\theta}_i$ such that the forward skinning function $\text{Skin}(\mathbf{X}^{\text{pers}}, \boldsymbol{\theta}_i, \boldsymbol{\pi})$ produces shapes as close as possible to $\mathbf{T}_i$. Because the function Skin is a complicated implicitly defined non-linear function, we introduce auxiliary variables $\mathbf{X}_i^{\text{arti}}$ for the personalized and articulated (posed) body shapes. When the inverse body modeling process is complete, we will have $\mathbf{X}_i^{\text{arti}} = \text{Skin}(\mathbf{X}^{\text{pers}}, \boldsymbol{\theta}_i, \boldsymbol{\pi})$, however, this equality does not have to hold in the intermediate steps of our optimization pipeline.

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

$$E_{\mathrm{targ}}(\mathbf{X}_i^{\mathrm{arti}}) = \sum_i \|\mathbf{N}_i^{\mathrm{T}}(\mathbf{S}^{\mathrm{skin}}\mathbf{X}_i^{\mathrm{arti}} - \mathbf{S}_i^{\mathrm{corsp}}\mathbf{T}_i)\|^2, \tag{4.8}$$

where $\mathbf{N}_i$ is a matrix of stacked scan normals, $\mathbf{S}^{\mathrm{skin}}$ is a binary selector matrix of surface vertices, and $\mathbf{S}_i^{\mathrm{corsp}}$ 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 $\mathbf{X}_i^{\mathrm{arti}}$ along their corresponding tangent planes at $\mathbf{T}_i$. The matrix $\mathbf{S}_i^{\mathrm{corsp}}$ is initialized to the identity (i.e., trusting the initial registration as described in Section 4.5.2) and after each iteration of the optimization process, we search for new correspondences. Specifically, for every skin vertex of $\mathbf{X}_i^{\mathrm{arti}}$, we search for closest point of $\mathbf{T}_i$, rejecting pairs further than 5 cm away or with normals differing by more than 30 degrees [RL01].

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

$$\begin{aligned} &\min_{\mathbf{X}^{\mathrm{pers}},\mathbf{X}_i^{\mathrm{arti}},\boldsymbol{\pi},\boldsymbol{\theta}_i} \; E_{\mathrm{targ}}(\mathbf{X}_i^{\mathrm{arti}}) + E_{\mathrm{reg}}(\mathbf{X}^{\mathrm{pers}},\boldsymbol{\pi}) \\ &\text{subject to } \nabla_{\mathbf{X}_i^{\mathrm{arti}}} E_{\mathrm{skin}}(\mathbf{X}^{\mathrm{pers}},\mathbf{X}_i^{\mathrm{arti}},\boldsymbol{\theta}_i,\boldsymbol{\pi}) = 0 \end{aligned} \tag{4.9}$$

where $i = 1\ldots\mathrm{numScans}$ as before. The equality constraints require the posed shapes $\mathbf{X}_i^{\mathrm{arti}}$ 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 $E_{\mathrm{reg}}(\mathbf{X}^{\mathrm{pers}},\boldsymbol{\pi})$. 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

$$\begin{aligned} E_{\mathrm{reg}}(\mathbf{X}^{\mathrm{pers}},\boldsymbol{\pi}) = \mathrm{BoneFlesh}(\mathbf{X}^{\mathrm{pers}},\boldsymbol{\theta}_0,\boldsymbol{\pi}) + E_{\mathrm{def}}(\mathbf{X}^{\mathrm{tmpl}},\mathbf{X}^{\mathrm{pers}}) \\ + w_{\mathrm{muscle}} E_{\mathrm{muscle}}(\mathbf{X}^{\mathrm{tmpl}},\mathbf{X}^{\mathrm{pers}}). \end{aligned} \tag{4.10}$$

Even though the sum of the BoneFlesh and E_{def} terms is reminiscent of the forward skinning function, here these terms have 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 $\mathrm{BoneFlesh}(\mathbf{X}^{\mathrm{pers}},\boldsymbol{\theta}_0,\boldsymbol{\pi})$ requires the personalized rest pose $\mathbf{X}^{\mathrm{pers}}$ to align with the skeleton grown according to skeletal growth 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 α parameters are free). The reason is the – perhaps optimistic – assumption that shape variations are more likely explained by muscle growth rather than by more general 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., assessment of body fat percentage by measuring the skin fold thickness. Note that there is no gravitational potential acting on $\mathbf{X}^{\text{pers}}$; it only acts on the final articulated shapes $\mathbf{X}_i^{\text{arti}}$. In other words, our $\mathbf{X}^{\text{pers}}$ shape corresponds to the rest-pose body in a zero gravity environment [CZXZ14], such as at the International Space Station.

Penalty method. Equation 4.9 represents a non-convex constrained optimization problem that can be written in a general form as $\min f(\mathbf{x})$ subject to $\mathbf{c}(\mathbf{x}) = 0$, where f is the objective and $\mathbf{c}$ a vector function of constraints. We solve this optimization problem by converting it into a sequence of unconstrained optimization problems using the penalty method [NW06]. Each unconstrained subproblem has the following form: $\min f(\mathbf{x}) + \gamma \|\mathbf{c}(\mathbf{x})\|^2$, where γ is the penalty weight. The γ parameter is progressively increased from 0 to 10^7 by factors of 10. (increasing the γ further does not produce any visible differences).

Each γ-subproblem is solved using Newton's method with Hessian modification (Algorithm 3.2 in [NW06]). In particular, evaluating the exact Hessian matrix would be complicated because it contains third derivative terms (note that the constraints $\mathbf{c}$ already contain first derivatives of the E_{skin} potential). Similarly to Bickel et al. [BKS*12], 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 Handling Collisions

We treat collisions in a fashion similar to McAdams et al. [MZS*11]. We detect tet-tet collisions using a fast bounding box sequence intersection algorithm [ZE00]. 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:

$$E_{\text{col}}(\mathbf{X}) = \left(\mathbf{n}_{''(\mathbf{X})}{}^{\top}(\mathbf{X} - {''}(\mathbf{X}))\right)^2, \tag{4.11}$$

where $''(\mathbf{X})$ is the projection of $\mathbf{X}$ onto the surface of the tetrahedral mesh, encoded by the barycenters of the closest surface triangle, and $\mathbf{n}_{''(\mathbf{X})}$ 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 [MZS*11]. The E_{col} energy potential is removed once the corresponding vertices are no longer in contact.

4.5.2 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 $\mathbf{T}_i$ is well aligned with its corresponding scan $\tilde{\mathbf{T}}_i$. 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. 4.5) 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 crude approximation is sufficient to establish good initial correspondences, which will be refined in subsequent iterations of our optimization process, as discussed earlier in the Targeting term paragraph.

4.6 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. The latter is particularly easy to achieve in our framework, because motion retargeting can be easily achieved using a subset of functionality of our optimization framework.

But first, let us explain how to introduce dynamics 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 $E_{\text{skin}}(\mathbf{X}^{\text{src}}, \mathbf{X}, \boldsymbol{\theta}_i, \boldsymbol{\pi})$ 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:

$$E_{\text{inert}}(\mathbf{X}) = \frac{1}{2h^2} \|\mathbf{M}^{1/2}(\mathbf{X} - 2\mathbf{X}_j^{\text{anim}} + \mathbf{X}_{j-1}^{\text{anim}})\|^2 \tag{4.12}$$

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 [BML*14].

In addition to the inertial term, we also add the collision avoidance potential E_{col} discussed in Section 4.5.1. 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. To do this, 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 [WET13], or calculated using inverse dynamics models [LST09]. 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. 4.6), however, instead of the rest-pose growth matrix $S(\alpha)$ (Eq. 4.7) we use the following matrix:

$$\mathbf{S}(\beta) = \begin{pmatrix} \beta^{-1} & 0 & 0 \\ 0 & \sqrt{\beta} & 0 \\ 0 & 0 & \sqrt{\beta} \end{pmatrix} \tag{4.13}$$

which accounts for the volume preserving nature of muscle contraction due to high water content in soft biological tissues [WMG96]. Mathematically, this is modeled by the fact that the determinant of matrix $\mathbf{S}(\beta)$ is one, resulting in the characteristic bulging behavior of contracting muscles (see Figure 4.11 for an example). Note that the muscle growth scaling matrix $\mathbf{S}(\alpha)$ (Eq. 4.7) does not have determinant one because it accounts for *growth*, which is of course not volume conserving.

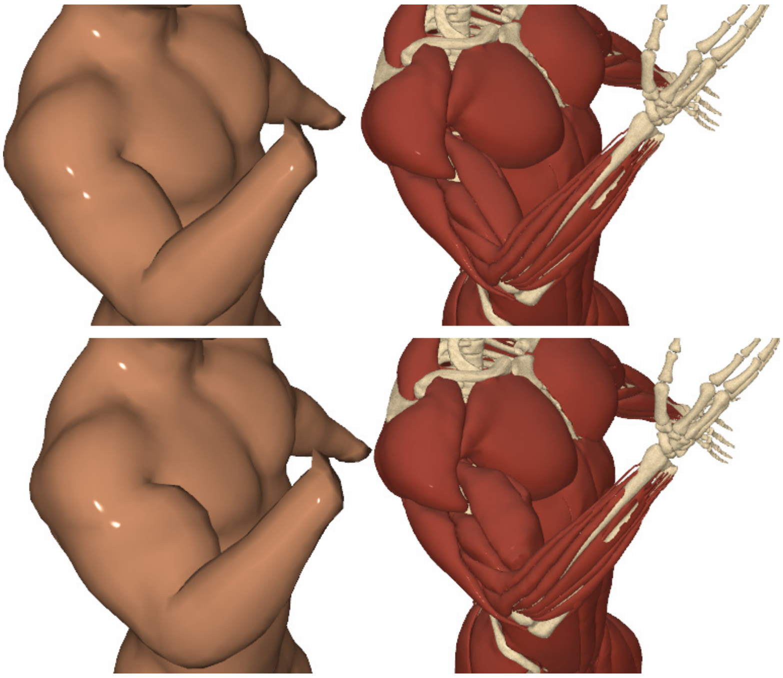


Figure 4.11 – Our physics-based animation approach allows for animating pose-specific muscle shape changes due to muscle contractions. The upper images show the shape of the arm and muscles in a flexing pose, and the lower images show the effect of contracting the biceps muscle in the same pose.

4.7 Results

We performed our experiments on 3D surface scans with diverse quality and resolution. Specifically, we tested our reconstructions on publicly avaiable good quality 3D surface scans obtained from the FAUST dataset [BRLB14] and database of Hasler et al. [HSS*09], from high quality commercially available scan collections and 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 4.12. We used between 2 to 5 scans for each subject depending on the quality of scans and diversity of the poses. Although it would possible to use only a single scan in our method (similarly to [DLG*13]), this means the underlying anatomical model is less well determined. In particular, 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

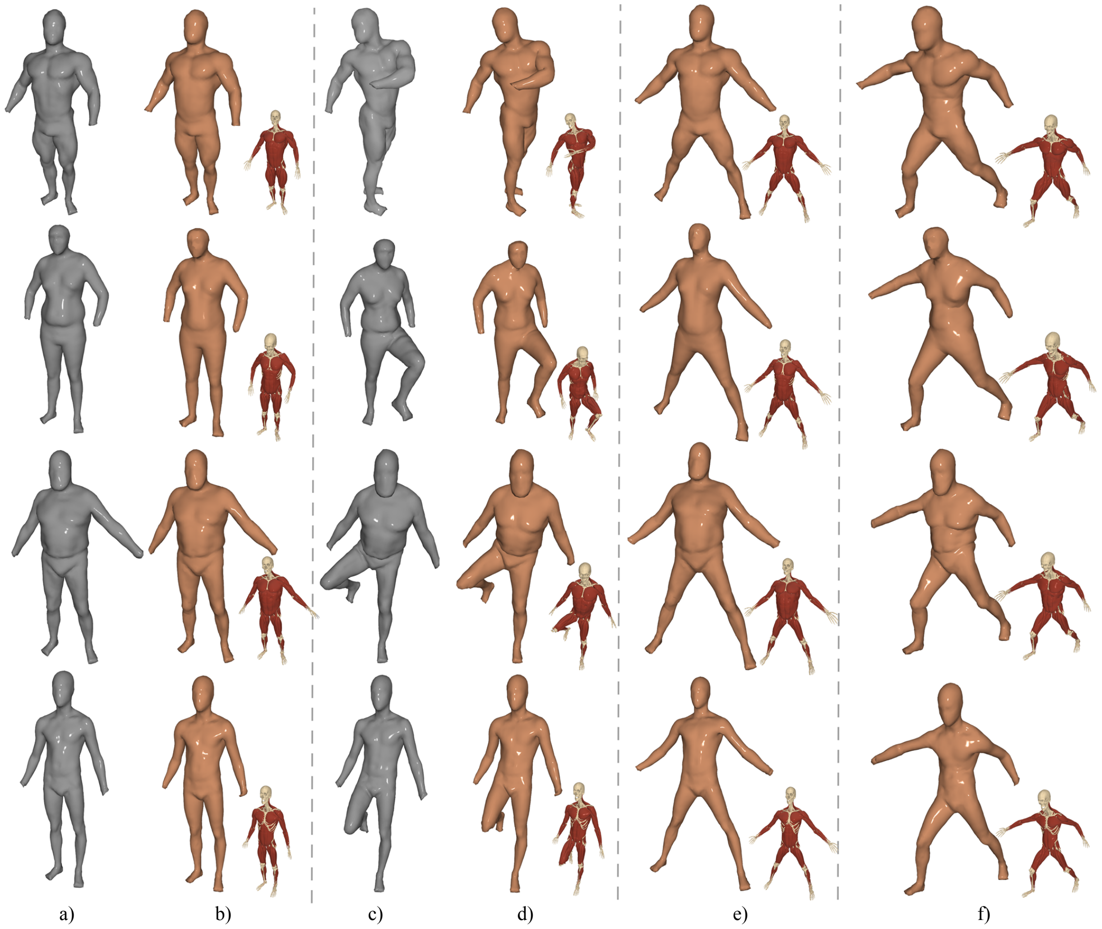


Figure 4.12 – Registered 3D surface scans of our test subjects in two different poses (a, c) and corresponding reconstructions using our anatomical physics-based model (b, d). Note that the shapes are quite similar. We also show our optimized rest pose $\mathbf{X}^{\text{pers}}$ (e) and a novel, unseen pose synthesized using our forward skinning model (f).

joints. Jointly optimizing over scans of multiple poses, e.g. adding a scan with bent limbs, helps to this eliminate uncertainty, as the optimization algorithm places the joint in the most appropriate location. In Figure 4.12 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.

Gravitational Effects Another advantage of using multiple scans is reducing the ambiguity due to gravitational effects and self-collisions of the skin. In Figure 4.13 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 [CZXZ14].

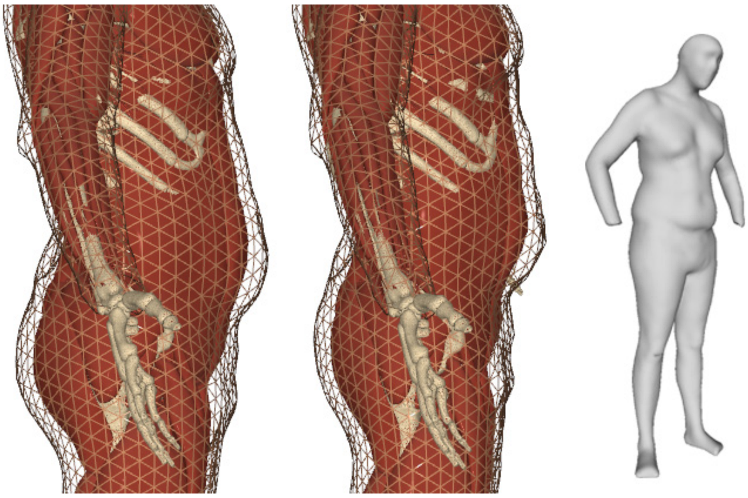


Figure 4.13 – Example of the effect of gravity on the rest pose reconstruction process. The figure on the left shows the result of the reconstruction without taking gravity into account. In the middle, gravity is taken into account and eliminated from the rest pose – note that the belly "floating" as 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.

Collisions An example of collision handling during the forward animation phase is shown in Figure 4.14. 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 4.15.

Comparison to Anatomy Transfer Our approach has several key advantages over Anatomy Transfer [DLG*13] and its more recent extensions [ZHK15]. First, our approach can take advantage of multiple scans in different poses, which leads to high reconstruction accuracy, as discussed above. Second, Anatomy Transfer as well as its extensions [ZHK15] use only a highly 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, as shown in Figure 4.16.

4.8 Implementation Details

The geometric search data structures and algorithms used for the scan registration and collision detection are based on CGAL [The16] and nanoflann [ML14]. Numerical linear algebra is

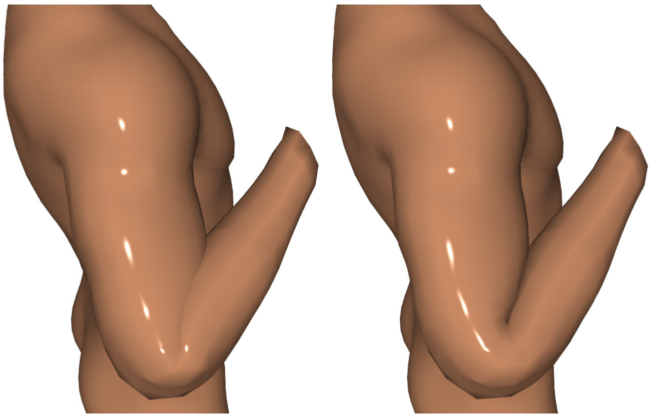


Figure 4.14 – Example of collision handling for the forward simulation.

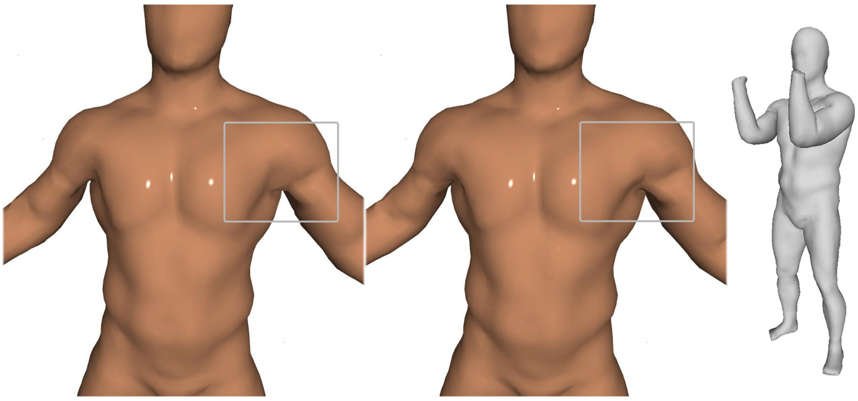


Figure 4.15 – Example of collision handling during inverse body modeling. In this example, a single scan was used (shown in gray) in which the actor was pressing his arms against his body. Notice that the rest pose reconstruction on the left has the shape of the arm imprinted on the chest; the rest pose reconstruction on the right does take the collision forces into account and reaches a more realistic rest shape.

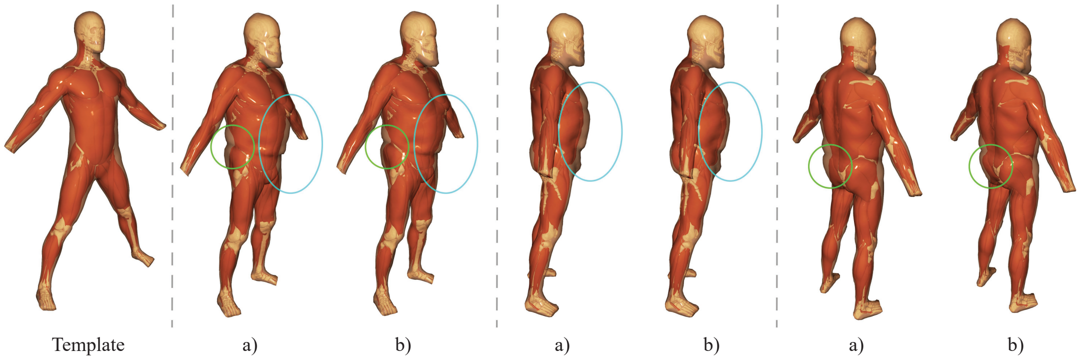


Figure 4.16 – Comparison of our method with muscle modeling and material-aware deformation (a) vs. uniform flesh deformation (b) similar to Anatomy Transfer

implemented using Eigen [GJ*10]. 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 paper 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: $k_{\text{bone}} = 10^{-1}$, $k_{\text{def\_bone}} = 7 * 10^{-4}$, $k_{\text{def\_tendon}} = 3 * 10^{-4}$, $k_{\text{def\_muscle}} = 2 * 10^{-4}$, $k_{\text{def\_soft\_tissue}} = 10^{-4}$, $k_{\text{muscle}} = 10^{-3}$.

4.9 Limitations and Future Work

We focus on capturing the physics of large- and medium-scale anatomical details, but we do not reconstruct faces, hands or toes. We believe that these are research topics of 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.

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 are added to the scans, e.g., by capturing a continuous stream of point clouds from an actor's performance. Furthermore, having the muscle forces actively change passive joint angles has been a topic intensely studied in the biomechanics community, but not tackled in great detail by Computer Graphics researchers.

Retrospective

A similar problem will be revisited in Chapter 6, but applied to the human head and specifically to modeling facial expressions. While at first sight the setting seems similar, animating human faces is fundamentally different. Skeletal muscles in the body contract and act on the skeletal joints, while facial muscles only deform the skin on the face (with the exception of the muscles of mastication). This makes it easier to build a coherent muscle-actuated facial template model.

In terms of optimization techniques, in our subsequent work we used the interior point method approach instead of the penalty method with Newton optimization. We found this to be superior for solving the inverse physics problem due to the improved solver, as well as the fact that we employed the IPOPT library which contains a mature implementation of this optimization algorithm.

Finally, Chapter 6 shows multiple applications of using physical constraints in order to produce realistic simulations of face modifications as a result gaining weight or facial surgery, for example. Such applications are also possible in the project described in this chapter, although we have not explored them yet.

5 Building and Animating User-Specific Volumetric Face Rigs

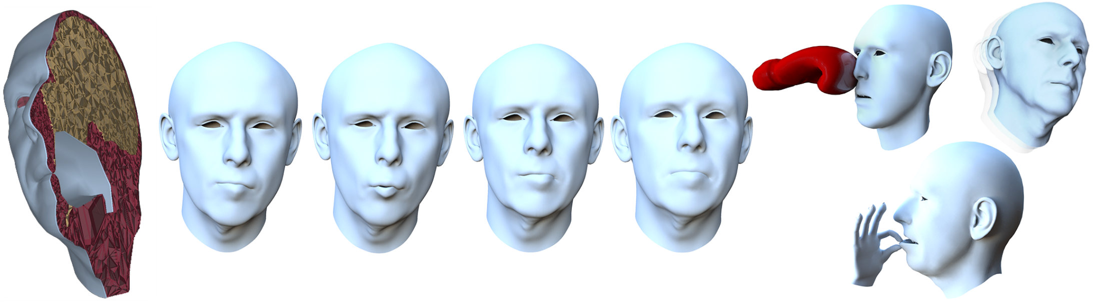


Figure 5.1 – We present a facial animation system that can simulate physics-based volumetric effects such as self-collisions and collision with external objects. Our method is data driven and avoids the burden of detailed anatomical modeling.

Note

This chapter corresponds to the following publication [IKNDP16]:

ICHIM, A.E., KAVAN, L., NIMIER-DAVID, M., AND PAULY, M. Building and Animating User-Specific Volumetric Face Rigs. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA), 2016*

The candidate contributed with most of the concepts and implementation in this publication.

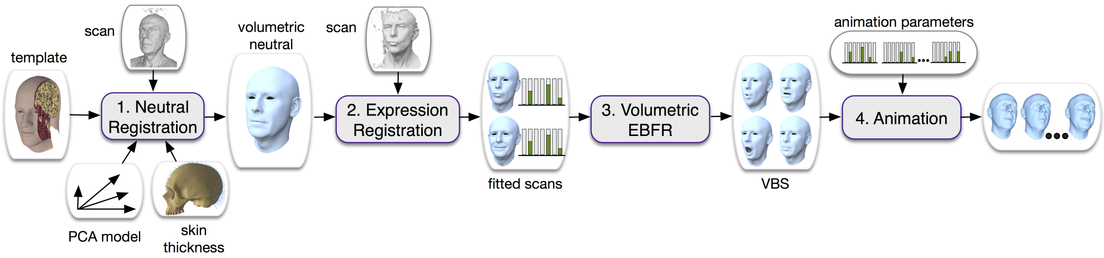


Figure 5.2 – Workflow of our method: from a template model and input 3D scans, our system produces a subject-specific facial animation model. We propose a volumetric formulation of example-based facial rigging (EBFR) to generate the volumetric blendshapes (VBS).

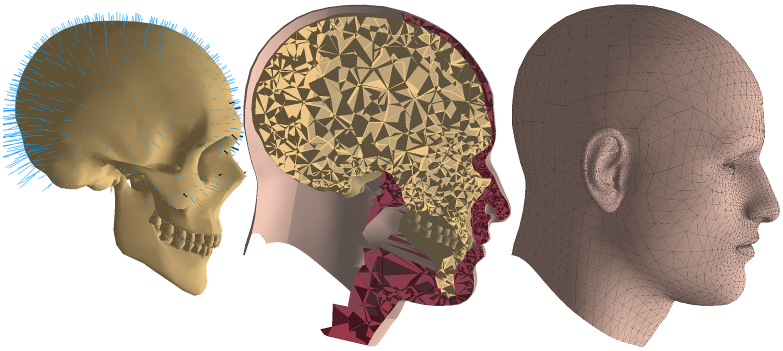


Figure 5.3 – Template model: skull of an average subject with expected flesh thicknesses (left), tet-mesh of the interior (middle), and skin (right).

Abstract

Currently, the two main approaches to realistic facial animation are 1) blendshape models and 2) physics-based simulation. Blendshapes are fast and directly controllable, but it is not easy to incorporate features such as dynamics, collision resolution, or incompressibility of the flesh. Physics-based methods can deliver these effects automatically, but modeling of muscles, bones, and other anatomical features of the face is difficult, and direct control over the resulting shape is lost. We propose a method that combines the benefits of blendshapes with the advantages of physics-based simulation. We acquire 3D scans of a given actor with various facial expressions and compute a set of *volumetric blendshapes* that are compatible with physics-based simulation, while accurately matching the input scans. Furthermore, our volumetric blendshapes are driven by the same weights as traditional blendshapes, which many users are familiar with. Our final facial rig is capable of delivering physics-based effects such as dynamics and secondary motion, collision response, and volume preservation without the burden of detailed anatomical modeling.

5.1 Introduction

Realistic animation of human faces is a long standing problem in computer graphics. Blendshape models are currently the most widely used solution in animation production [LAR*14] and impressive facial animations have been created with blendshape models in recent high-end productions. However, this process can be very labor-intensive and time-consuming even for experienced digital artists. Physics-based simulation of anatomically-based face models can potentially eliminate much of this manual work, because non-linear effects such as incompressibility of biological soft tissues or prevention of self-collisions (e.g. lips-lips or lips-teeth) can be handled automatically. However, the anatomy of the human face is highly complex, posing significant difficulties in creating accurate anatomical face models of specific people.

Instead, we explore a new route, proposing a facial animation model that leverages the benefits of physics-based simulation without the burden and complexity of full anatomical modeling. Specifically, our technique helps prevent geometric inconsistencies such as volume loss, inter-penetrations, or unnatural facial expressions commonly observed in traditional blendshape models. Even though these deficiencies can be manually fixed by a skilled artist using corrective blendshapes, our method achieves physically-realistic behavior automatically, without the need of user intervention.

Our goal is to build an animatable facial rig of a specific actor. We start by acquiring 3D scans of several facial expressions of the actor including a neutral face shape. These scans are used to adapt a volumetric head template, corresponding to an average human (see Figure 5.3), to the specific actor. To achieve physics-based behavior, we propose a novel *volumetric blendshape* model, which controls the deformation gradients in the entire face volume.

The proposed volumetric blendshapes model retains the key desirable properties of traditional blendshapes: posing with intuitive blendshape weights and direct control over the resulting deformations. This means that any animator familiar with traditional blendshape models will be able to readily use our method. In contrast to traditional blendshapes, our model performs a full physics-based simulation, allowing even effects such as inertia or collisions with external objects. This is enabled by the fact that our volumetric blendshapes control deformation gradients of the flesh instead of absolute positions. However, we do not model individual muscles, which would require significant modeling effort and simulation time. Instead, the volumetric blendshapes discretize the entire deformable volume of the face using a tetrahedral mesh.

Our method (see Figure 5.2) assumes an average-human volumetric head model as input. To create an actor-specific face model, we scan the actor in a neutral pose and several (in the order of 10) facial expressions. In the first step, *Neutral Registration* in Figure 5.2, we volumetrically warp the template to align with the input scan of the actor's neutral facial expression. In step 2, we perform *Expression Registration* to deform this neutral shape into the acquired facial expressions, such as smile, frown, etc. The key difference from the first step is that now we assume the bones are rigid and the soft tissues are incompressible, because at this stage we do not model a new human being, but rather explain different facial expressions of the same actor. Due to the fact that our models are volumetric, we obtain full volumetric deformation for each of the facial expressions.

In order to create a facial rig compatible with traditional blendshape models, step 3: *Volumetric EBFR* executes a volumetric version of Example-Based Facial Rigging [LWP10], i.e., explaining each of the expression scans using a blend of volumetric blendshapes. The key idea of volumetric blendshapes is to perform non-linear blending of deformation gradients of all tetrahedra in our face model. On one hand, volumetric blendshapes are driven by the same weights as traditional blendshapes, constituting a convenient interface for the *Animation* stage of our pipeline. On the other hand, volumetric blendshapes approximate muscle contraction forces, i.e., the generators of facial expressions. This allows us to combine them with other competing forces in a physics-based simulation, enabling us to deliver effects such as secondary motion and inertia, volume preservation, and contact forces.

Contributions. We present a pipeline to turn 3D scans of an actor's face into physics-based simulation-ready models that are able to respond to inertia or external forces, e.g., due to self-collisions of the face or collisions with external objects. We formulate our pipeline in a coherent optimization framework – all components are built using the concepts of Projective Dynamics [BML*14], which 1) results in efficient run times and 2) can be easily reproduced using open source implementations of Projective Dynamics such as ShapeOp [DDB*15]. Several novel technical contributions make this approach practically viable: 1) novel registration methods using physics-based priors such as volume preservation and self-collision handling, 2) advanced collision handling for Projective Dynamics, and 3) a "baking" system for generating higher-order corrective blendshapes which explain physical effects such as volume preservation and collisions with performance comparable to traditional blendshapes.

In this paper we focus on creating simulation-ready volumetric models. We do not aim for complete production-quality facial rigs that are commonly equipped with high resolution textures, normal, or displacement maps, see Figure 5.4. Compared to traditional blendshape

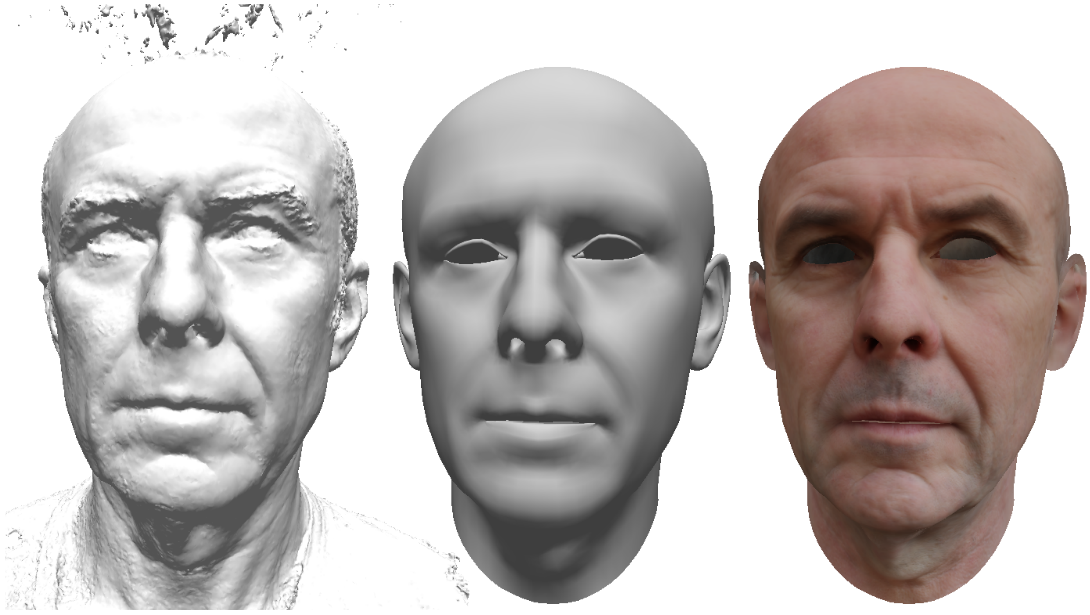


Figure 5.4 – Input hi-res 3D scan (left). Our volumetric physics-based model (middle) uses only a medium-resolution mesh, but details can be re-introduced using high-resolution textures (right), as is commonly done in high-end productions.

models, our approach provides more accurate volume and area preservation, as well as rigid motion of the skull and the jaw. Our model also handles interactions between the lips and the teeth, often prone to self-intersections with traditional blendshape models, in particular for speech or chewing sequences. We can also simulate interactions with external objects, e.g., responding to contacts with rigid bodies.

5.2 Related Work

Facial reconstruction. Research in the field of facial animation has mostly focused on data-driven techniques, due to the high complexity of facial morphology. The seminal work of [BV99] builds a statistical (PCA) model of facial geometry and later on [CWZ*14] builds a bilinear facial model, which can be employed to create blendshape models from a single image [BV99], [CWZ*14], from multiview stereo [ABF*07], [ARL*10], or for the creation of personalized real-time tracking profiles from RGB-D data [WBLP11], [BWP13] or monocular video [IBP15],[GVWT13], [SWTC14].

Anatomical models. Dicko et al. [DLG*13] propose a method for transferring and editing the internal structure of human bodies. They use a template human body model containing the skeleton and internal organs and register it to new surface-mesh humanoid models. The exterior surfaces are registered and the internal volume is adapted using harmonic deformation. Additional constraints are used for manually tuning the amount of fat tissue and keeping the bones straight. In a similar vein, [ZHK15] adapts the bone structure of upper and lower limbs given an RGB-D sequence of moving limbs. [CBB*15] propose a technique to transfer facial anatomy to challenging non-human creatures using sophisticated correspondences between the template and target shapes. However, their method relies only on a single neutral facial expression. In contrast, our approach uses multiple scans of facial expressions and is able to reproduce them with high accuracy.

[VCL*06] present a review of computerized techniques for craniofacial reconstruction, i.e., generating the skin surface of faces from 3D skull information. An algorithm to reconstruct the skin surface, as well as an animatable muscle system from 3D scans of skulls is proposed by [KHS03]. Their method registers a template face model to the 3D mesh of the skull by RBF deformation on a sparse set of landmarks with user-specified skin thicknesses. A mass-spring system is then adapted to the fitted template and the face can be animated. For more application-specific use cases of anatomical models, [BB14] present an approach for rigid stabilization of the head in high quality 3D scans by fitting a simple skull model with physically-inspired constraints. [BBK*15] use high quality facial scanning and a simplified physical model in order to recover spatio-temporal details of the eyelids.

Physics-based facial animation. [SNF05] build a system for physics-based animation of one human subject. The subject's face is captured using a laser scanner (high-resolution, surface only) and an MRI scanner (low-resolution, volumetric). A simulation-ready 3D model is created using custom software tools, medical atlases, and multiple months of manual work. The resulting face model is biomechanically accurate in the sense that realistic facial expressions are created by physics-based simulation of muscle activations. In addition, the model can be used to track a facial performance of the subject, captured using a sparse set of markers attached to the face. The physics simulator is based on a quasi-static FEM approach, numerically solved using Newton's method.

More recent techniques such as Position-based [MHHR07] and Projective Dynamics [LBOK13, BML*14] propose to substitute Newton's method with faster numerical solution procedures. In particular, Projective Dynamics [BML*14] yields faster per-iteration times while simultaneously enjoying high robustness and support of many different types of deformation constraints.

Combining simulation and data. Our volumetric blendshapes blend deformation gradients, similarly to MeshIK [SZGP05]. However, MeshIK relies only on deformation gradients of surface triangles and does not support dynamics or collisions. Similar approaches such as deformation transfer [SP04] and FaceShift [WBLP11] also do not take collisions into account, see Figure 5.6. We use a complete volumetric model combined with full physics-based simulation, enabling us to deliver inertial and secondary motion effects (such as flesh jiggling) as well as realistic response to collisions while preserving the volume of biological soft tissues. [MWF*12] build a mass-spring system model for the face that is able to deliver some of these effects. However, volume preservation with mass-spring systems is problematic. A concurrent work [BSC16] uses Projective Dynamics to deform the surface of a face combined with a new concept of "blendforces", which are similar to our volumetric blendshapes. However, [BSC16] model only the surface of the face. In contrast, our method explicitly models volume preservation of the flesh, as well as rigidity of the skull and the jaw bones.

5.3 Method

As input, we assume a *template* model of an average human face. This model consists of a volumetric tetrahedral mesh for the neutral expression which discretizes the interior of the head, including a realistic model of the oral cavity, see Figure 5.3. We obtain this model by converting a commercial anatomical CAD model of the head [Zyg16] into a tet-mesh using the method of [JKSH13]. The *skin* is the boundary of this tet-mesh. To get an initial model of facial deformations, we use an artist-created surface blendshape model [WBLP11], which also comes with parameterization (UV coordinates). We register this model against the boundary of our volumetric model, which allows us to animate the skin, but not the interior. Extending the surface deformations to the interior is one aspect of our pipeline, discussed below.

Our final volumetric template model is a single connected tet-mesh where we can identify the following components corresponding to high-level anatomical features of the head (see Figure 5.3): 1) skin – a UV-mapped surface mesh, 2) bones – tet-meshes for the cranium and the mandible, including teeth, 3) flesh – in-between tet-mesh conforming to the skin and the boundaries of the bones.

Our volumetric model corresponds to a hypothetical average human subject and must be adapted to a given actor. The scanning of our actor's face is performed using a custom multiview stereo rig with 12 DSLR cameras with uniform lighting, similar to [BBB*10]. Note that our method is not dependent on the specific scanning method. Any approach for creating high-resolution scans of a face, e.g. laser scanning, RGB-D, are equally suitable. The captured photos are processed in AgiSoft PhotoScan which creates detailed triangle meshes for each expression.

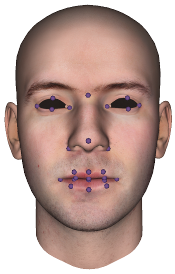
5.3.1 Volumetric modeling of actor's neutral face

Registration. The 3D scan of the actor's neutral face is a triangle mesh containing noise, topological errors, and other imperfections, see Figure 5.7. We overcome these issues by regularized registration, i.e., by deforming our volumetric template model to align well with the 3D scan of the actor. We follow the paradigm of Iterative Closest Point (ICP) algorithms and iterate between finding correspondences and volumetric deformations of our template. We find surface correspondences using the standard approach of closest points with distance and normal-based rejection [RL01]. The non-rigid deformation steps are alternated with shape-preserving rigid fitting steps, which only allow for translation, rotation, and uniform scale (necessary because multi-view stereo does not determine scale).

Deformation model. We model volumetric deformations in the Projective Dynamics framework due to its speed, robustness, and flexibility [BML*14]. The key concept of Projective Dynamics is to use elastic energy potentials expressed in the following "projective" form:

$$E_i(\mathbf{x}) = \|\mathbf{G}_i\mathbf{x} - P_i(\mathbf{G}_i\mathbf{x})\|_F^2, \tag{5.1}$$

where E_i is the energy contribution due to element number i (e.g., tetrahedron), $\mathbf{x}$ is a column vector concatenating all of the nodal coordinates (deformed state), $\mathbf{G}_i$ is a sparse matrix, typically representing a discrete differential operator, and P_i is a projection operator. For example, the finite element As-Rigid-As-Possible model (E_i^{ARAP}) [CPSS10] can be expressed with $\mathbf{G}_i$ representing the deformation gradient of a tetrahedron [SB12] and P_i representing the projection onto $SO(3)$, i.e., the group of 3D rotations.



Correspondence terms. Our registration process utilizes a set of 26 landmark correspondences initialized automatically using [SLC11] and fine-tuned by the user (see the figure on the right). In the Projective Dynamics framework, these correspondences are implemented using an "attachment" term E_i^{attach} where $\mathbf{G}_i$ is simply a selector matrix and P_i is the constant target position (i.e., projection onto a fixed point). The correspondences found through closest point search by the ICP algorithm are handled similarly; the only difference is that we do not “trust” the absolute positions of these correspondences and therefore use a point-to-plane energy term $E_i^{\text{planeDist}}$, where $\mathbf{G}_i$ is still a selector, but P_i projects on the plane tangent to the scan at the closest point. This allows for tangential sliding, which improves the convergence of the ICP process [LSP08]. The point-to-plane energy is also used as a collision response mechanism, projecting inter-penetrated vertices outside of the volume; we elaborate on collision processing in Section 5.3.5.

Face priors. We also add energy terms specific to faces, i.e., utilizing the prior knowledge that the resulting surface must correspond to a plausible human face. As we are solving for deformations of the interior too, ideally we would also use a statistical shape model of skulls. However, so far we were not successful in obtaining a sufficiently large database of 3D skull shapes. Instead, we utilize flesh thickness measurements from a forensic study [DGCV*06], inspired by the work of Beeler and Bradley [BB14] on rigid stabilization.

Statistical shape models of neutral faces of various people are available; we use the established PCA model of Blanz and Vetter [BV99]. This model consists of a mean face shape $\mathbf{m}$ and 50 PCA basis vectors, represented as orthonormal columns of a matrix $\mathbf{B}$. Each of the basis vectors is associated with a standard deviation, represented as a 50×50 diagonal matrix Σ. Let us also denote by $\mathbf{S}$ a surface selector matrix, i.e., $\mathbf{Sx}$ represents the boundary (skin) vertices, discarding the interior ones. The skin shape $\mathbf{Sx}$ can be additively decomposed into two parts: one in the column space of $\mathbf{B}$ and the other one orthogonal to it. We introduce a different energy term for each part. For the component of $\mathbf{Sx}$ in the column space of $\mathbf{B}$ we can measure its likelihood of corresponding to a natural face shape, as predicted by our PCA model. This leads to $E^{\text{faceLike}}(\mathbf{x}) = \|\Sigma^{-1/2}\mathbf{B}^{\mathsf{T}}(\mathbf{Sx}-\mathbf{m})\|^2$. The orthogonal complement $(\mathbf{I}-\mathbf{BB}^{\mathsf{T}})(\mathbf{Sx}-\mathbf{m})$ corresponds to modes outside of our PCA model. We do not have standard deviations for these modes and therefore we penalize them uniformly using the term $E^{\text{faceDist}}(\mathbf{x}) = \|(\mathbf{I}-\mathbf{BB}^{\mathsf{T}})(\mathbf{Sx}-\mathbf{m})\|^2$. Both of these terms are convex quadratic functions that can be easily embedded in the Projective Dynamics framework.

Flesh thickness. Our flesh thickness model is based on statistical information from a forensic study [DGCV*06]. We start from a sparse set of 16 skull landmarks containing the mean and variance of flesh thickness at this point, and then linearly interpolate these values over the entire skull. Specifically, for each non-landmark skull vertex, we find three closest landmarks, with closeness measured using geodesic distance on the skull. The mean and variance are then interpolated linearly, using the inverse geodesic distances as blending weights. The resulting mean thicknesses are visualized in Figure 5.3 (left). Regions such as the craniocervical junction and the teeth do not have flesh thickness measurements (in these regions, we set the mean to zero and the standard deviation to infinity). For each skull vertex j, we introduce an energy term:

$$E_j^{\text{thickness}}(\mathbf{x}) = \frac{1}{\sigma_j^2} \|\mathbf{n}_j^{\mathrm{T}}(\mathbf{H}_j\mathbf{x} - \mathbf{T}_j\mathbf{x}) - \mu_j\|^2 \tag{5.2}$$

where σ_j is the standard deviation, $\mathbf{n}_j$ is the skull normal, $\mathbf{H}_j$ is the selector of the skull vertex and $\mathbf{T}_j$ selector of the corresponding skin vertex, and μ_j is the mean flesh thickness. The term $E_j^{\text{thickness}}(\mathbf{x})$ encourages realistic placement of the skull inside the head, see Figure 5.5. We combine all of the face-specific priors into:

$$E^{\text{prior}} = E^{\text{faceLike}} + E^{\text{faceDist}} + \tau \sum_j E_j^{\text{thickness}} \tag{5.3}$$

For notational brevity we drop the argument $\mathbf{x}$ which appears in all the terms. The parameter $\tau \geq 0$ expresses the relative confidence in the flesh thickness prior.

For a given set of correspondences, the final volumetric deformation problem can be expressed as the minimization of:

$$E^{\text{total}} = E^{\text{planeDist}} + \alpha E^{\text{attach}} + \beta E^{\text{ARAP}} + \gamma E^{\text{prior}}, \tag{5.4}$$

where we assume that each energy type is summed over all elements, e.g., $E^{\text{ARAP}}(\mathbf{x}) = \sum_i E_i^{\text{ARAP}}(\mathbf{x})$, with i summing over all tetrahedra. The weights $\alpha \geq 0, \beta \geq 0, \gamma \geq 0$ are used to guide the registration process. The key idea is to start with high regularization (high values of α, β, γ) to obtain an initial guess and progressively reduce the regularization as our correspondences are becoming more and more accurate. Specific parameter values used in our experiments can be found in Section 5.5.

In terms of numerical optimization, we minimize E^{total} using the local/global solver of Projective Dynamics [BML*14]. We slightly modify the solver in order to handle constraints using Lagrange multipliers, which allows us to avoid collision constraints in a more efficient way, as described in Section 5.3.5. We denote the final result as $\mathbf{x}_{\text{neutral}}$, see the third column of Figure 5.7.

5.3.2 Registration of actor's facial expressions

In the previous section we showed how to deform the volumetric template into $\mathbf{x}_{\text{neutral}}$, which corresponds to the scan of our actor in neutral expression. In this section, we describe how to

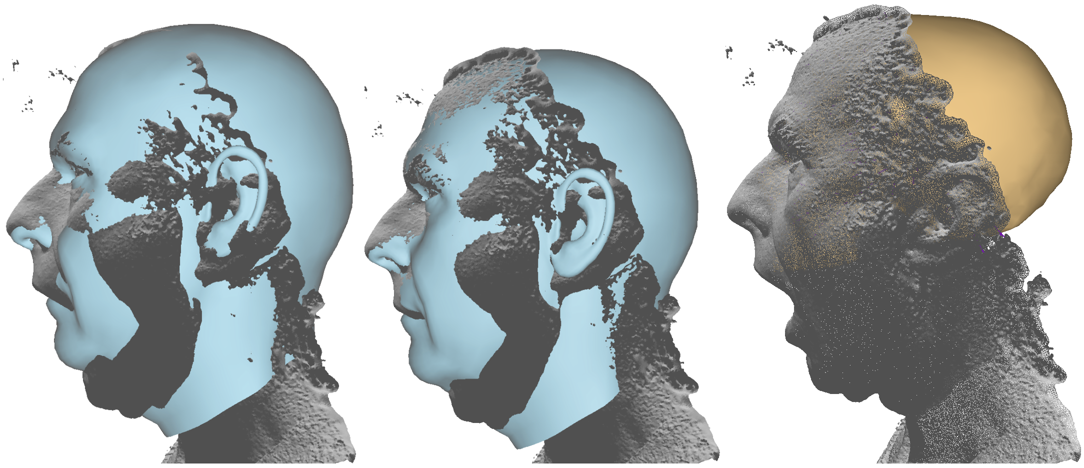


Figure 5.5 – Rigid stabilization using the skull mesh and skin thicknesses. The standard skin registration approach (left) does not compute the correct rigid registration of a mouth open scan, as compared to the skull-based approach (middle and right).

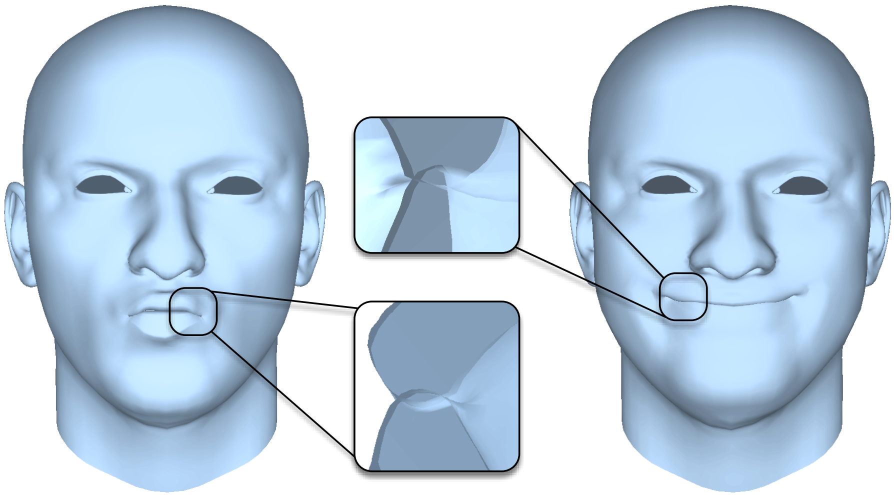
deform $\mathbf{x}_{\text{neutral}}$ to align with the other expression scans. Specifically, we use 10 expressions such as smile, frown, kiss, sneer, etc. The key difference from the previous section is that the deformation from $\mathbf{x}_{\text{neutral}}$ to the target expression must be physiologically plausible, i.e., achievable by a normal human subject under normal conditions. For example, in Section 5.3.1 it is accepted to deform the bones, because we are explaining individual subject-specific differences. However, in the next stage the bones must remain rigid, because now we are explaining only shape differences due to facial motion of a given human subject.

For each facial expression of our actor (Figure 5.7) we manually find approximate corresponding blendshape weights. This is not too difficult because the actors were instructed to assume specific expressions, which are combinations of only a few blendshapes. We use deformation transfer [SP04] to bootstrap the expression registration process. Assuming a given facial expression, for each triangle of the template surface mesh (2D), we compute the deformation gradient, i.e., the 3D linear transformation between the rest pose and the template expression, using the cross product of the edges to determine the normal, as in [SP04]. Next, we select all surface tetrahedra from the neutral pose ($\mathbf{x}_{\text{neutral}}$) and define an energy term

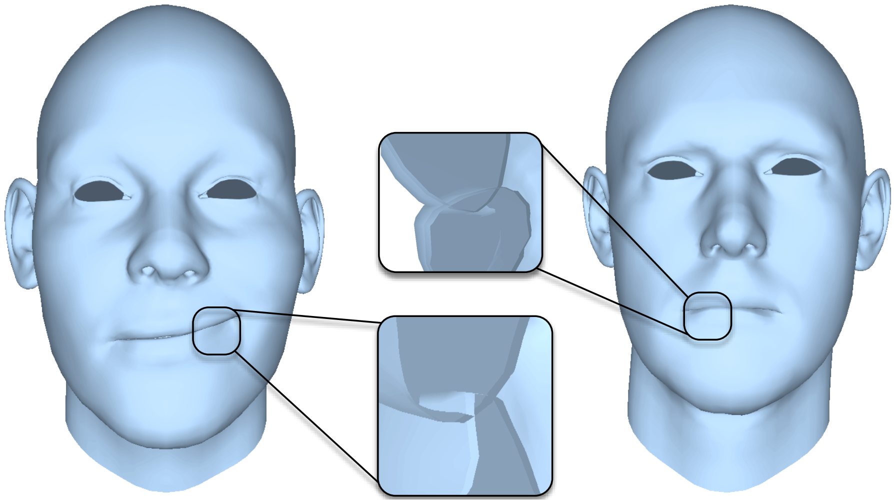
$$E_k^{\text{defTransfer}} = \|\mathbf{F}_k - \mathbf{F}_k^{\text{target}}\|_F^2, \tag{5.5}$$

which attracts the deformation gradients $\mathbf{F}_k$ of all surface tets k of the neutral face ($\mathbf{x}_{\text{neutral}}$) to the deformation gradients $\mathbf{F}_k^{\text{target}}$ calculated from the template model.

Because the template blendshape model explains only the surface, the terms $E_k^{\text{defTransfer}}$ are defined only for tetrahedra adjacent to the boundary. To propagate the surface deformation to



(a) FaceShift [WBLP11] scan registration.



(b) Deformation transfer [SP04].

Figure 5.6 – Most previous methods do not handle self-collisions.

the entire volumetric shape, we apply the E^{ARAP} term discussed in Section 5.3.1 to all of the tets. This term ensures that the surface deformation is distributed throughout the entire volume. During this volumetric deformation, we need to account for the fact that most biological soft tissues are nearly incompressible [WMG96]. We capture this behavior with a new term E^{volume} that is analogous to the ARAP term, except that the projection on $SO(3)$ is replaced with projection of $SL(3)$ – the group of matrices with determinant 1, i.e., volume preserving linear maps. This leads to the objective

$$E^{\text{defTransfer}} + \mu E^{\text{ARAP}} + \lambda E^{\text{volume}}, \tag{5.6}$$

where the μ and λ are Lamé parameters approximating the elasticity of the flesh. We minimize Equation 5.6 using Projective Dynamics, keeping the vertices corresponding to the bones fixed (they do not appear as degrees of freedom in the optimization problem). We open the jaw manually by estimating the rigid transformation of the jaw corresponding to the given expression. We denote the result as $\mathbf{x}_{\text{init}}$, which serves as volumetric initialization for the subsequent fitting.

Next, we need to take the actual expression scan into account. As shown by Beeler and Bradley [BB14], it is advantageous to start the fitting process with “rigid stabilization”, guided by areas of the skin that are close to the skull and thus not significantly affected by facial expressions. We use an energy analogous to Equation 5.2, where the mean is set to the actual flesh thickness in $\mathbf{x}_{\text{neutral}}$ and the variance is left out, because at this point we are no longer trying to model variations among different human subjects. We denote this modified objective as $\tilde{E}^{\text{thickness}}$. We find the optimal transformation $\mathbf{T}$ as a composition of rotation, translation, and uniform scale such that $\tilde{E}^{\text{thickness}}(\mathbf{T}\mathbf{x}_{\text{init}})$ is minimized. The uniform scale takes care of the fact that the expression scan from multi-view stereo is in arbitrary units of length.

The resulting “rigidly stabilized” state $\mathbf{T}\mathbf{x}_{\text{init}}$ contains a good estimate of the bone positions and a good initialization of the skin. We are therefore ready to launch the ICP process to account for the subtleties of flesh deformations, while keeping the bones fixed. The deformation energy is analogous to Equation 5.4:

$$E^{\text{exp-total}} = E^{\text{planeDist}} + \alpha E^{\text{attach}} + \mu E^{\text{ARAP}} + \lambda E^{\text{volume}} \tag{5.7}$$

Similarly to Section 5.3.1, the attachment term E^{attach} is found in a semi-automatic way using [SLC11]. Differently from Equation 5.4, we drop the E^{prior} term because at this stage we are already committed to a given actor. For the same reason, we include the E^{volume} term to enforce incompressibility of the soft tissues.

5.3.3 Volumetric facial rigging

The expression registration process described in Section 5.3.2 results in plausible volumetric shapes $\mathbf{x}_{\text{expression},l}$, where l indexes the individual facial expressions. Interpreting $\mathbf{x}_{\text{neutral}}$ (Section 5.3.1) as the rest pose, we can compute deformation gradients for all tets, mapping from $\mathbf{x}_{\text{neutral}}$ to $\mathbf{x}_{\text{expression},l}$. For each expression, we stack the deformation gradients of all tets into a matrix $\mathbf{H}_l$. Let us denote the vector of blendshape weights for the l-th expression as α_l. These blendshape weights are copied from the template blendshapes and ensure that our volumetric blendshapes will have the same semantics as the template blendshapes. This has the desired consequence that the user intuitively understands how each parameter affects the shape of the face, e.g., that $\alpha_{l,6}$ lowers the right mouth corner etc.

Our next task is to find the volumetric blendshapes. A volumetric blendshape is a collection of deformation gradients for all tets in the face model. Even in the traditional surface case [LWP10], we do not observe the blendshapes directly, because each facial expression $\mathbf{x}_{\text{expression},l}$ is composed of several blendshapes. We find our volumetric blendshapes through a process similar to Example-based Facial Rigging [LWP10] adapted to the volumetric case. Specifically, we solve for volumetric blendshapes $\mathbf{V}_m$ by minimizing:

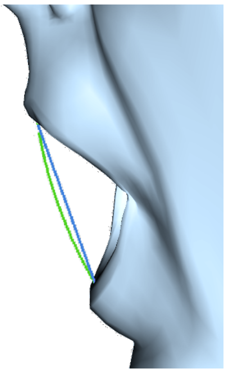
$$\sum_l \left|\left|\left(\mathbf{I} + \sum_m \mathbf{V}_m \alpha_{l,m}\right) - \mathbf{H}_l\right|\right|_F^2 + \kappa \sum_m \|\mathbf{V}_m - \tilde{\mathbf{V}}_m\|_F^2 \tag{5.8}$$

where the addition of stacked identity matrices $\mathbf{I}$ ensures that if all $\alpha_{l,m} = 0$, we obtain the neutral face, corresponding to all deformation gradients equal to identities. In other words, the $\alpha_{l,m}$ are not coefficients of an affine combination, but rather scaling factors of individual blendshapes, interpreted as differences from the neutral pose. In the second term, the $\tilde{\mathbf{V}}_m$ are volumetric blendshapes obtained from deformation transfer of template blendshapes, i.e., minimizing Equation 5.6. The second term including its weighting coefficient $\kappa \geq 0$ expresses a prior, which is necessary because the first (data) term does not specify the volumetric blendshapes uniquely (in all of our experiments we use $\kappa = 10^{-4}$). This is because we use only a small set of expressions which could be generated by many different volumetric blendshapes. Therefore, we use the second (regularization) term that picks a unique solution – the one that is as close as possible to deformation-transferred template blendshapes.

5.3.4 Animation

We create new facial animations using a time-varying sequence of blendshape weights $\mathbf{w}(t)$ and rigid head motion $\mathbf{R}(t) \in SE(3)$; the latter specifies the position and orientation of the skull. Even though the jaw motion could be also controlled explicitly, we continue to rely on the blendshape model, which is compatible with standard animation workflows, i.e., the jaw motion is implicitly controlled via blendshape weights instead of explicit control via rigid transformations or a kinematic rig (used by Sifakis and colleagues [SNF05]). The rigidity of the jaw bone will be enforced in the volumetric-blendshape blending process, described below. Our input sequences of the time-varying **w** and **R** parameters can be either directly keyframed by artists or captured from human subjects using tracking software such as FaceShift [WBLP11].

The blendshape weights can be used to blend the deformation gradients from the individual volumetric blendshapes linearly, $\mathbf{F}^{\text{target}} = \mathbf{I} + \sum_m \mathbf{V}_m \alpha_m$, as in Equation 5.8. However, it is a well-known fact that linear blending of matrices is prone to artifacts, especially when the blended transformations contain larger rotations [SD92]. This problem can be avoided by using the polar decomposition method introduced by Shoemake and Duff [SD92]. Specifically, if we have a set of 3×3 matrices $\mathbf{M}_1, \ldots, \mathbf{M}_n$, we first find their polar decompositions, i.e., $\mathbf{M}_i = \mathbf{R}_i \mathbf{S}_i$, where $\mathbf{R}_i$ is a rotation and $\mathbf{S}_i$ is symmetric. The rotations $\mathbf{R}_i$ are then blended non-linearly using quaternions [Sho85]; the "stretch" matrices $\mathbf{S}_i$ are blended linearly, as they correspond to the non-rigid component of the transformation. Finally, the blended rotations and stretch components are multiplied together to create the final result. This approach avoids the loss of volume associated with linear blending of rotations. If the input transformations are pure rotations, as is the case for tets corresponding to the jaw, the blended result will also be a pure rotation, guaranteeing that the jaw bone remains rigid as expected. See the figure on the right for an example: the blue curve is the path of a linearly interpolated vertex for a mouth opening sequence, while the green curve is the path using nonlinear interpolation.



In theory, Equation 5.8 should be revised for polar decomposition-based blending. In practice, the computation of polar decomposition inside the objective would require more complicated numerical solution procedures and therefore, we continue to rely on Equation 5.8. This linear approximation seems to be sufficient for the purpose of determining volumetric blendshapes.

If we denote the deformation gradients computed by polar-decomposition-blending as $\mathbf{F}^{\text{target}}$, we can create a "targeting" energy term:

$$E^{\text{target}}(\mathbf{x}) = \|\mathbf{F}(\mathbf{x}) - \mathbf{F}^{\text{target}}\|_F^2 \tag{5.9}$$

where **F** is a linear function of **x** [SB12]. This energy specifies that all deformation gradients **F** of the unknown mesh state **x** are attracted to $\mathbf{F}^{\text{target}}$. Intuitively speaking, the E^{target} term serves the same purpose as muscle activations in full anatomical models [SNF05], however, without the need of modeling the geometry and mechanics of individual muscles. While we avoid the intricacies of full anatomical modeling, we retain the possibility of introducing additional energy potentials and constraints. For example, dynamic effects can be easily added using an "inertial" term $E^{\text{inertia}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{y})^{\top}\mathbf{M}(\mathbf{x}-\mathbf{y})$, where $\mathbf{M}$ is the mass matrix and $\mathbf{y}$ is state predicted by Newton's first law, i.e., motion without the presence of forces. This term is equivalent to the variational Implicit Euler formulation used in Projective Dynamics [BML*14]. Perhaps even more useful is the ability to add constraints due to collisions with the face itself, e.g., lips-lips or lips-teeth collisions, or external objects. Our approach to handling contact involves a modification of the Projective Dynamics solver which is described in the following section.

Stronger inertial or contact forces can result in shapes with deformation gradients significantly departing from the targeting term E^{target}. In order to preserve realistic behavior of the soft tissues even in these large deformations, we add the $\mu E^{\text{ARAP}} + \lambda E^{\text{volume}}$ terms, as in Equation 5.6. This has a natural biomechanic interpretation as the elasticity of passive soft tissues [TSIF05]. Intuitively, if there is, e.g., a large external force acting on the cheek, this force is propagated through the entire musculoskeletal system. For tets corresponding to the skull and the jaw, we use stiffness high enough to prevent any visible deformations of the bones (specifically, we use $\mu = 1000$).

5.3.5 Collisions

Our collision processing mechanism is based on point-to-plane constraints which are dynamically instanced as needed to resolve collisions, analogous to classical collision resolution approaches [MZS*11]. To detect inter-penetrations, we use a fast bounding box sequence intersection algorithm [ZE00] for the broad phase, and an AABB tree built in the rest pose. For efficiency, only certain pairs of regions of the face are checked against collisions (e.g., lips against lips, lips against skull, skin against external objects). When colliding with external objects, our current implementation assumes these external objects are fixed, e.g., directly controlled via keyframing. In either case, if we detect a collision, i.e., a vertex penetrating a tetrahedron, we find the closest surface point where the vertex needs to move in order to resolve the collision. To facilitate sliding, we create a constraint which requires the offending vertex to align with a tangent plane at the closest surface point. In case of both self-collisions and external collisions, this can be expressed as affine equality constraint $\mathbf{C}_i\mathbf{x} = \mathbf{d}_i$, where i indexes contact points. We append all of the collision constraints together: $\mathbf{C}\mathbf{x} = \mathbf{d}$. The main challenge in efficient collision processing is the fact that the collision constraints $\mathbf{C}\mathbf{x} = \mathbf{d}$ are

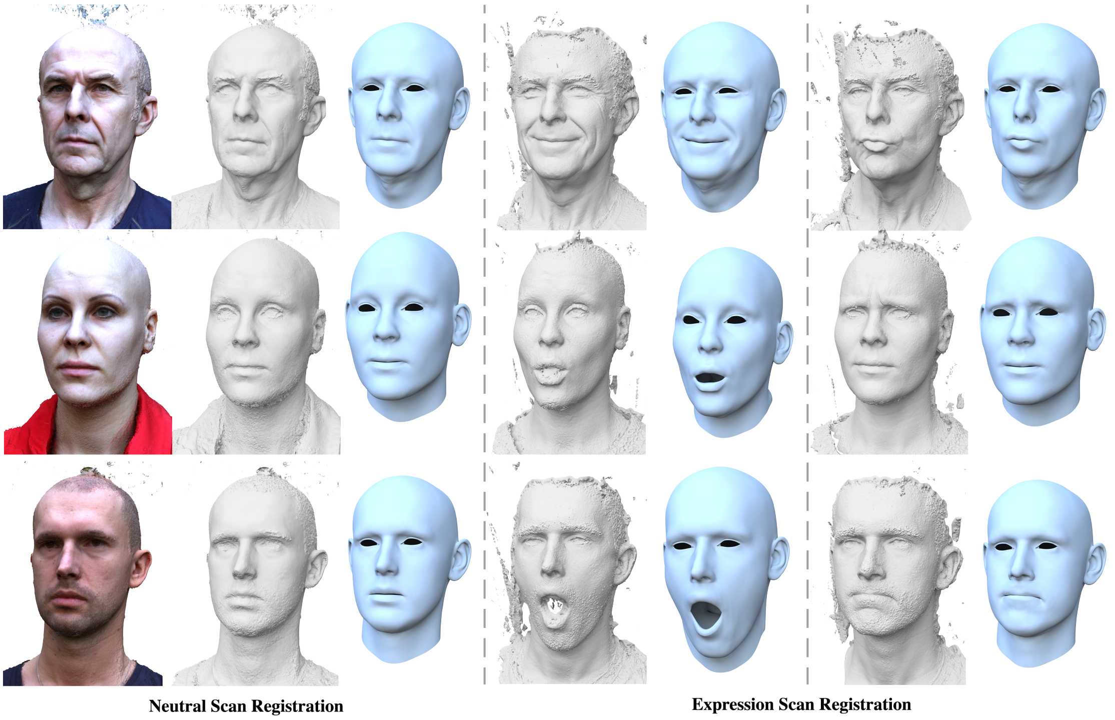


Figure 5.7 – Registration of 3D scans of our test subjects: neutral pose (left) and two facial expressions (middle, right).

frequently changing.

The original Projective Dynamics paper [BML*14] proposes two options. The first is to directly add energy terms penalizing violation of the collision constraints. Unfortunately, this requires re-computing the factorization of the global step matrix, resulting in significant computational overheads. The second option is to add these constraints for all vertices in the system and pre-factorize only once, because changing the target positions or planes of the constraints affects only the right hand sides. The undesired side-effect is that these constraints affect the behavior of the system even if there are no collisions. The collision constraints are always present in the system, and even if they are not active, they attract the vertices towards their current locations. In practice, this introduces additional damping, slowing down convergence in the quasi-static case and creating artificial viscosity in the dynamic case.

To avoid these drawbacks, we propose a new method, motivated by the observation that the number of colliding vertices is typically small, because the collision resolution process is invoked each iteration. The key idea is to apply the Schur complement [Jac13, YCP16] to reuse the pre-computed factorization without introducing any artificial damping. First, recall that the global step of Projective Dynamics solves a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A}$ is a constant symmetric positive definite matrix. Therefore, Projective Dynamics pre-computes a sparse

Cholesky factorization of $\mathbf{A}$ that allows calculating $\mathbf{A}^{-1}\mathbf{b}$ very efficiently as long as $\mathbf{A}$ is not changing.

We propose to incorporate our frequently changing collision constraints $\mathbf{Cx} = \mathbf{d}$ using Lagrange multipliers. This leads to the KKT system, named after the famous Karush-Kuhn-Tucker optimality conditions [NW06]:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix} \tag{5.10}$$

One possible way to solve this system while taking advantage of the existing factorization of $\mathbf{A}$ would be using low-rank updates [CDHR08]. Unfortunately, in our case the cost of low-rank updates is comparable or even greater to the cost of factorizing the KKT system from scratch. Instead, we propose to solve for the Lagrange multipliers using the Schur complement of Equation 5.10: $\mathbf{CA}^{-1}\mathbf{C}^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{CA}^{-1}\mathbf{b} - \mathbf{d}$. The matrix $\mathbf{CA}^{-1}\mathbf{C}^{\mathsf{T}}$ is dense but small, because we assume the number of rows of $\mathbf{C}$ is small; in our simulations, it is typically less than 50. The solve for $\boldsymbol{\lambda}$ is therefore fast even with dense linear algebra. Having found $\boldsymbol{\lambda}$, we can compute the solution $\mathbf{x} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{C}^{\mathsf{T}}\boldsymbol{\lambda})$.

5.4 Corrective blendshapes

In some cases, physics-based facial animation may not be desirable, e.g., in 3D game engines which require extremely fast animation algorithms. In this case, our approach can be used as an automatic method to generate corrective blendshapes, which is a common way to address the problems of linear blendshape models [LAR*14]. We focus on the basic case of quadratic blendshapes, even though higher-order methods are also possible. The key idea is to sample activations of every pair of blendshapes. For each pair, we sample activations of each of the two blendshapes; we use four steps for the first weight: $0.25, 0.5, 0.75, 1$ and five for the second one: $0, 0.25, 0.5, 0.75, 1$, leading to a total of 20 samples per pair. We denote the final sequence of $20\binom{b}{2}$ blendshape weights samples as $\mathbf{w}_1, \mathbf{w}_2, \ldots$, where the number of blendshapes in our case is $b = 29$. For each of them we synthesize a realistic face shape using our method, as described in Section 5.3, and denote the coordinates of the resulting skin vertices as $\mathbf{p}_1, \mathbf{p}_2, \ldots$. Our goal is to explain these example face shapes $\mathbf{p}_k$ using the quadratic blendshape model. This task can be formulated as an optimization problem:

$$\operatorname*{arg\,min}_{\mathbf{m}, \mathbf{u}_i, \mathbf{v}_{ij}} \sum_k \left\| \mathbf{m} + \sum_i w_{k,i}\mathbf{u}_i + \sum_i \sum_j w_{k,i} w_{k,j} \mathbf{v}_{ij} - \mathbf{p}_k \right\|^2 \tag{5.11}$$

where $\mathbf{m}$ is the mean, corresponding to neutral facial expression, $\mathbf{u}_i$ are traditional linear blendshapes and $\mathbf{v}_{ij}$ are the quadratic blendshapes. We find the optimal $\mathbf{m}, \mathbf{u}_i, \mathbf{v}_{ij}$ by solving a linear least squares problem.

5.5 Implementation and results

The geometric search data structures and algorithms used for registration and collision detection are based on CGAL. Our optimization framework is an extension of the open-source ShapeOp [DDB*15]. Numerical linear algebra is handled using Eigen. Our current prototype runs on the CPU, parallelized using OpenMP. We benchmark the performance on a consumer laptop with a 2.5 GHz Intel Core i7 processor and 16GB of main memory. In our experiments, the animation converged using 6 iterations per frame. The timing per frame ranges from 500ms if no collisions are detected up to 1200ms when the lips collide heavily (about 80 collision constraints at a time, like in the *chewing* sequences shown in the supplementary video). The template volumetric model has 7366 vertices and 14600 triangles for the skin surface, 8947 vertices and 36654 tetrahedra for the flesh, 6760 vertices and 29888 tetrahedra for the bones. We use the same anatomical template for all of our actors.

Registration. For registration of the neutral face expression (Section 5.3.1), we used the following parameters: $\alpha = 10^1, \beta = 10^1, \gamma = 10^{-2}, \tau = 10^1$. We captured three different human subjects, all of them experienced actors. The input neutral scans and our resulting registered templates are shown in Figure 5.7 (left). In addition to the neutral expression, for each actor we also captured 10 facial expressions and executed the expression fitting algorithm described in Section 5.3.2 with parameters $\mu = 10^2$ and $\lambda = 10^3$. The results for two different expressions can be seen in Figure 5.7 (middle and right). Our registration technique takes advantage of collision constraints to avoid self-penetrations, see Figure 5.8. Similarly, the volume preservation terms used in the expression registration process help us avoid unnatural deformations, as shown in Figure 5.9. Because the inside of the mouth is not visible and therefore not captured by 3D scanning methods, previous techniques that do not account for incompressibility of the flesh can deform the lips into unnaturally thin shapes. Furthermore, volume preservation helps to establish the lip contact surface, which is difficult to determine using optical methods due to occlusions.

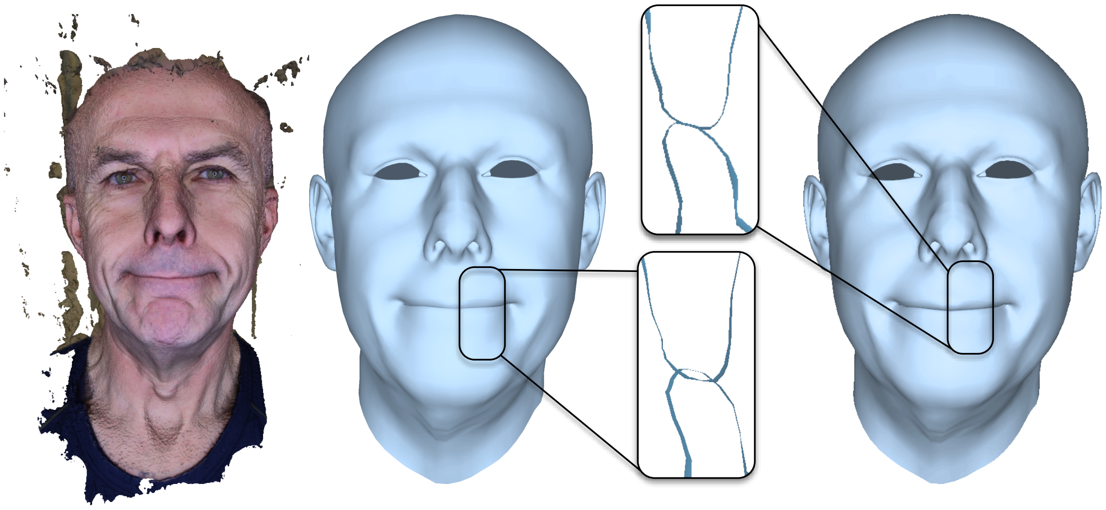


Figure 5.8 – Our collision handling (right) avoids inter-penetrations during expression registration.

Animation. We invite the reader to watch the accompanying video, showing facial animation sequences generated by our system. In particular, certain types of facial expressions frequently produce self-intersections of the lips with traditional blendshape models. Our method successfully removes these inter-penetrations while departing from the original blendshape model as little as possible, see Figure 5.10.

In addition to traditional facial motion driven purely by muscle activations, our method allows incorporating external forces. In Figure 5.11 (left), as well as in the accompanying video, we show a talking sequence with part of the bottom lip held fixed. Our simulator can also naturally deliver dynamic effects, including stylized animations such as shockwave propagation through the skin or making the nose more heavy while swinging the head, see Figure 5.11 (middle). Perhaps even more entertaining are collisions with external objects, such as the boxer glove in Figure 5.11 (right). Note that the nose bridge does not deform due to the presence of the bone in this region, unlike the rest of the nose.

Corrective blendshapes. We use 8120 samples corresponding to activating all pairs of blendshapes at different activation levels (Section 5.4), resulting in 406 quadratic blendshapes which require additional 65MB of memory (in addition to 7.7MB for the linear blendshapes). The runtime increases from 1ms for linear-only blendshapes to 8ms, which is acceptable even in real-time applications such as games. To compare the accuracy of quadratic vs. linear blendshapes, we measured for each frame of an animation sequence the error between the full simulated model and an approximation computed by 1) linear and 2) quadratic blendshapes.

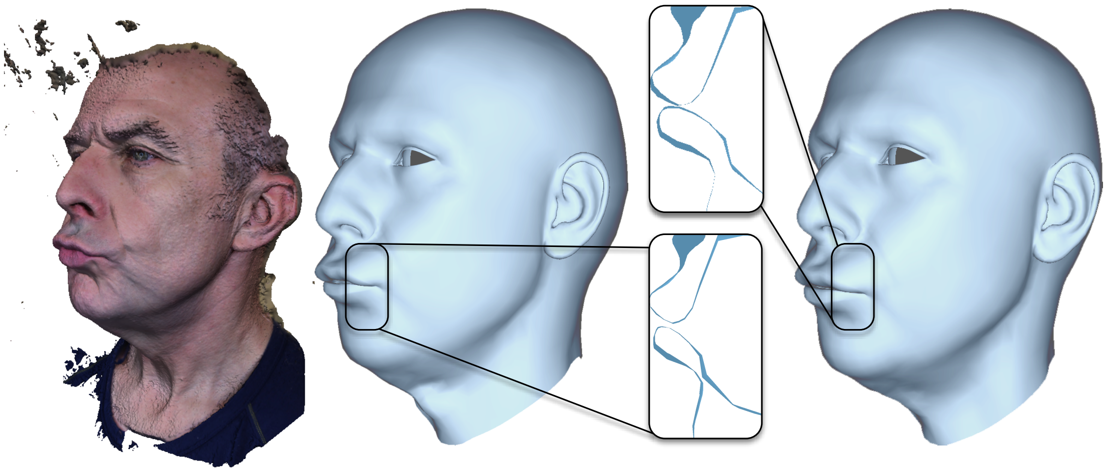


Figure 5.9 – Volume preservation allows us to achieve more natural expression registration (right). To the left is the result without volume preservation.

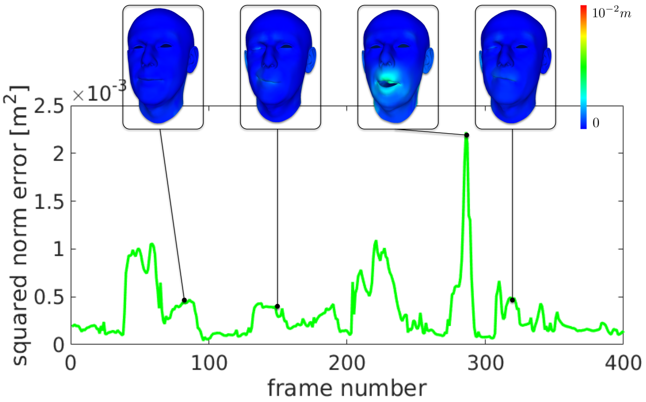


Figure 5.10 – The difference between the blendshape animation and our physically simulated animation, expressed as the squared norm error between each mesh for each frame of a sequence. Note that the spikes appear when large non-linear motion is present (e.g., frame 280), or when collisions are present (e.g. frames 90, 155, 330).

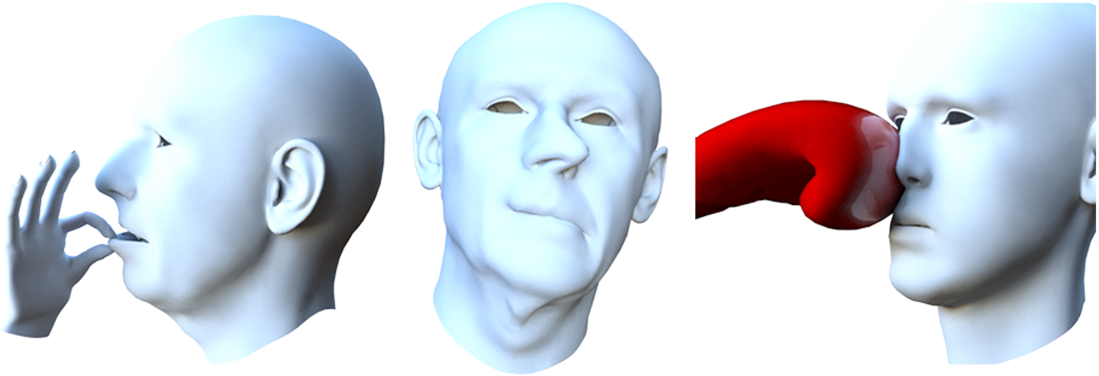


Figure 5.11 – Our method allows us to incorporate external forces and dynamic effects.

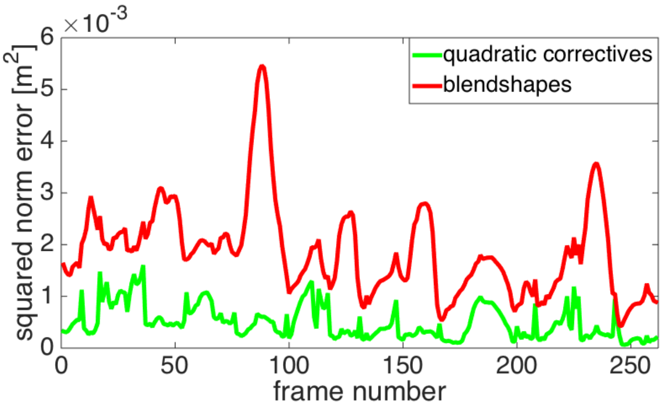


Figure 5.12 – Error decrease when using blendshapes against our trained quadratic corrective blendshapes on an animation sequence.

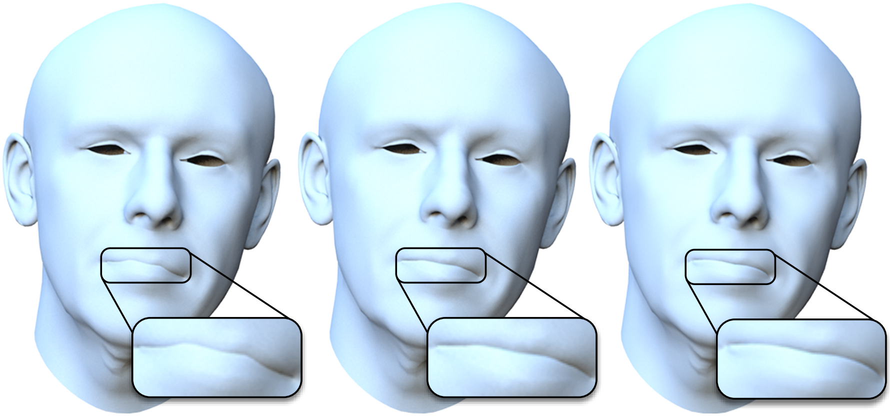


Figure 5.13 – An example of the handling of self-collisions via corrective blendshapes. From left to right: linear blendshapes, quadratic correctives, simulation.

The resulting plot is shown in Figure 5.12. The quadratic blendshapes significantly reduce the error compared to the linear ones. Even though we cannot guarantee collision-free results, the quadratic blendshape model is quite effective in avoiding visible self-penetrations, as demonstrated in Figure 5.13. A limitation of quadratic blendshapes is the fact that they are not able to capture previously unseen external forces, such as collisions with external objects.

5.6 Conclusion

We introduced a method for creating personalized volumetric face rigs that combine the intuitive control of blendshapes with the improved realism of physics-based simulation. Specifically, our face animation supports volume preservation, avoids self-collisions, and enables dynamic effects due to external forces. These improvements in animation quality come at the cost of increased computation time. To alleviate this performance loss, we show how the simulated face model can be used to automatically create corrective blendshapes. While these cannot guarantee the same level of accuracy as the full simulation model, significant quality improvements are achieved with a low computational overhead compared to the initial blendshape model.

Building a volumetric face rig based on high-resolution surface scans requires advanced registration algorithms to mitigate errors caused by the inherent limitations of the optical 3D scanning process, such as occlusions. We show how the same underlying optimization framework used for animation can be applied effectively for volumetric registration as well. This unification of representation and optimization leads to a simple and robust implementation based on existing open-source software.

As the quest for more realism continues, we believe that reducing the complexity of facial rigging will be crucial for wide-spread adoption in computer gaming, movie production, VR and avatar-based online communication. Interesting future challenges lie in further simplifications of the acquisition process, in building more advanced volumetric priors for effective model reconstruction, and in more efficient simulation methods for realtime animation of volumetric face rigs.

5.7 Acknowledgements

We thank the anonymous reviewers for their feedback and constructive criticism. We would also like to thank Sofien Bouaziz, Matthew Cong, Ron Fedkiw, Eftychios Sifakis, and Peter Shirley for valuable discussions and feedback. This project was supported in part by NSF awards IIS-1622360 and IIS-1350330 and a gift from Activision. Furthermore, we would love to acknowledge the help received from the actors who accepted to be scanned for the purpose of this project: Peter Ender, Jördis Wölk, and Michael Schönert, as well as Anton Rey for the coordination and acting advice.

Retrospective

While the approach presented in this chapter enabled us to produce numerous compelling animations, we did notice some limitations that we have addressed in more detail in the next chapter. The major drawback was the way we constrain the tetrahedra when performing expressions. Because we impose the rotational component for each tetrahedron (Equation 5.9), we essentially have a linear elasticity model, which we will show to cause visual artifacts under large rotations due to heavy inertia or collisions. Second, from a mathematical point of view, the essential property of the system to be in a steady state in the scans is not respected in this approach, and we will correct that with the inverse physics solution in Chapter 6. Furthemore, the jaw motion will be modeled as an actual rigid body, creating boundary constraints in the physics solves. These additions, as well as others, made the forward animation more robust and lively, allowing for a large range of novel result applications.

However, we will use this project as the *anatomy transfer* registration module for the next chapter, as we have shown it performs much better than thin shels deformation models for facial scan registration.

6 Phace: Physics-based Face Modeling and Animation

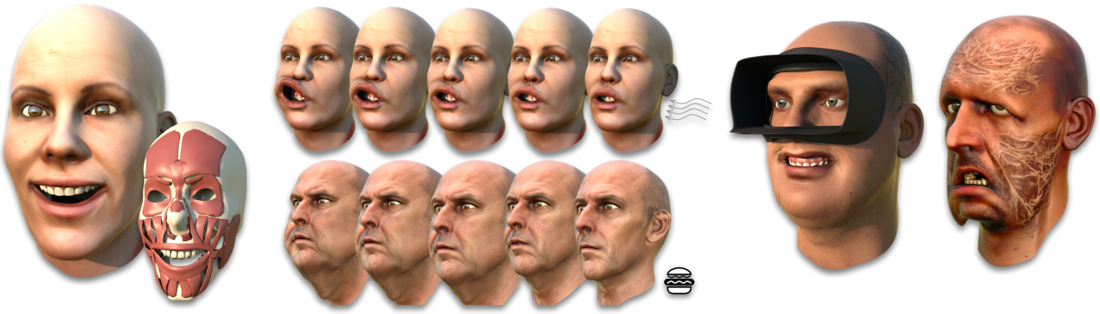


Figure 6.1 – Physics-based simulation facilitates a number of advanced effects for facial animation, such as applying wind forces, fattening and slimming of the face, wearing a VR headset, and even turning into a zombie.

Note

This chapter corresponds to the following publication [IBP15]:

Ichim, A.E., Kadlecek, P., Kavan, L., and Pauly, M. Phace: Physics-based Face Modeling and Animation, *ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2017*

The candidate contributed as follows:

- collecting the data
- the inverse physics formulation of the muscle activation model and its regularization
- some of the applications
- the implementation of the registration and animation pipelines

Abstract

We present a novel physics-based approach to facial animation. Contrary to commonly used generative methods, our solution computes facial expressions by minimizing a set of non-linear potential energies that model the physical interaction of passive flesh, active muscles, and rigid bone structures. By integrating collision and contact handling into the simulation, our algorithm avoids inconsistent poses commonly observed in generative methods such as blendshape rigs. A novel muscle activation model leads to a robust optimization that faithfully reproduces complex facial articulations. We show how person-specific simulation models can be built from a few expression scans with a minimal data acquisition process and an almost entirely automated processing pipeline. Our method supports temporal dynamics due to inertia or external forces, incorporates skin sliding to avoid unnatural stretching, and offers full control of the simulation parameters, which enables a variety of advanced animation effects. For example, slimming or fattening the face is achieved by simply scaling the volume of the soft tissue elements. We show a series of application demos, including artistic editing of the animation model, simulation of corrective facial surgery, or dynamic interaction with external forces and objects.

6.1 Introduction

Accurate simulation of facial motion is of paramount importance in computer animation for feature films and games, but also in medical applications such as regenerative and plastic surgery. Realistic facial animation has seen significant progress in recent years, largely due to novel algorithms for face tracking and improvements in acquisition technology [vdPJD*14, KRP*15].

High-end facial animations are most commonly produced using a sophisticated data capture procedure in combination with algorithmic and manual data processing. While video-realistic animations can be created in this manner, the production effort is significant and costly. A main reason is that complex physical interactions are difficult to recreate with the commonly employed reduced model representations. For example in blendshape rigs, collisions around the lip regions or inertial effects of the facial tissue are typically not accounted for. To remedy these shortcomings, artists often introduce hundreds of corrective shapes that need to be carefully sculpted and blended to achieve the desired effect in each specific animation sequence [LAR*14].

Recent work [IKNDP16, BSC16] proposes to avoid these shortcomings by augmenting the generative approach of blendshape animation with a simulation-based solution. A key benefit of physics-based simulation is the ability to correctly handle collision and contact, both for internal contact of facial tissue or bones, as well as for collisions with external objects. In addition, secondary motion, such as inertial deformations or other time-dependent effects can easily be integrated into the optimization pipeline.

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 [BY86]. Accurate simulation requires building a detailed volumetric face model that faithfully represents the shape and dynamics of the captured subject. However, acquiring such a volumetric face model is challenging. Volumetric data produced by CT or MRI scanners is often difficult to convert into a simulation-ready representation. So far, successful pioneering methods required a significant amount of manual editing [SNF05], which makes them difficult to deploy at scale.

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 6.3). By scanning the subject in multiple facial poses, we obtain a representation of the geometry and expression dynamics of the acquired person. 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.

We propose a novel muscle activation model in order to match the input scans more accurately. Unlike previous models that are constrained by fixed fiber directions, our model introduces additional degrees of freedom to support any deformation devoid of global rotation (since a muscle cannot rotate itself). This generalized model avoids the problem of relying on pre-determined fiber directions which are often inaccurate.

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.

Contributions. The main technical contributions of our work are:

- a novel muscle activation model that offers more flexibility than standard fiber-based

models,
- 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 inverse modeling optimization to adapt the simulation template to a series of expression scans of a specific person.

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 paper 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.

Overview. Figure 6.2 provides a visual summary of our physics-based face modeling and animation approach. Central to our method is a face template model that combines volumetric and surface elements as shown in Figure 6.3. 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 provide 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,

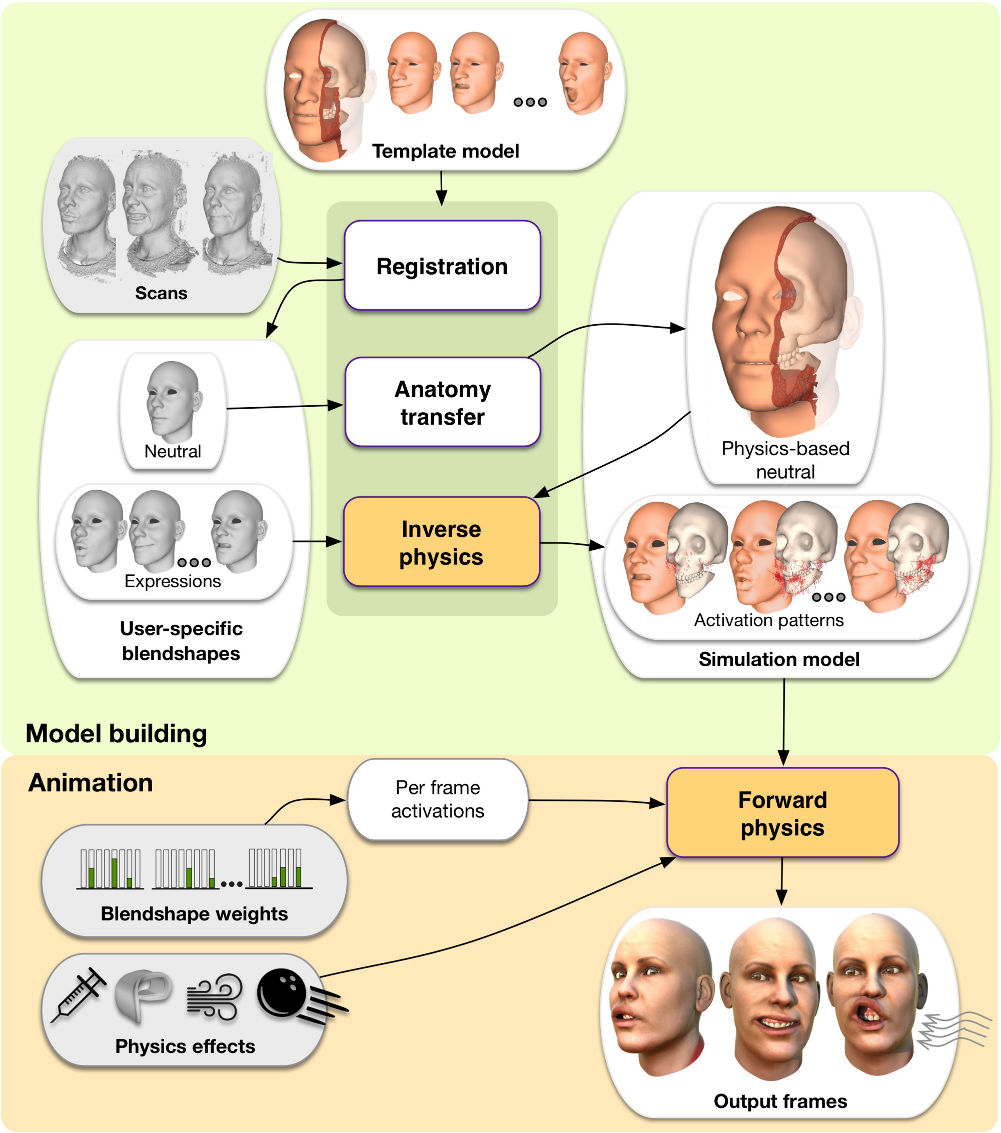


Figure 6.2 – Schematic workflow of our method.

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.

The rest of the paper is organized as follows: we first put our work in context by discussing related work in Section 6.2. In Section 6.3 we present our simulation template model. Then we introduce the forward and inverse physics simulation algorithms in Sections 6.4 and 6.5, respectively. Section 6.6 explains how these components are integrated into the model building and animation stages. In Section 6.7 we analyze the behavior of our method and provide

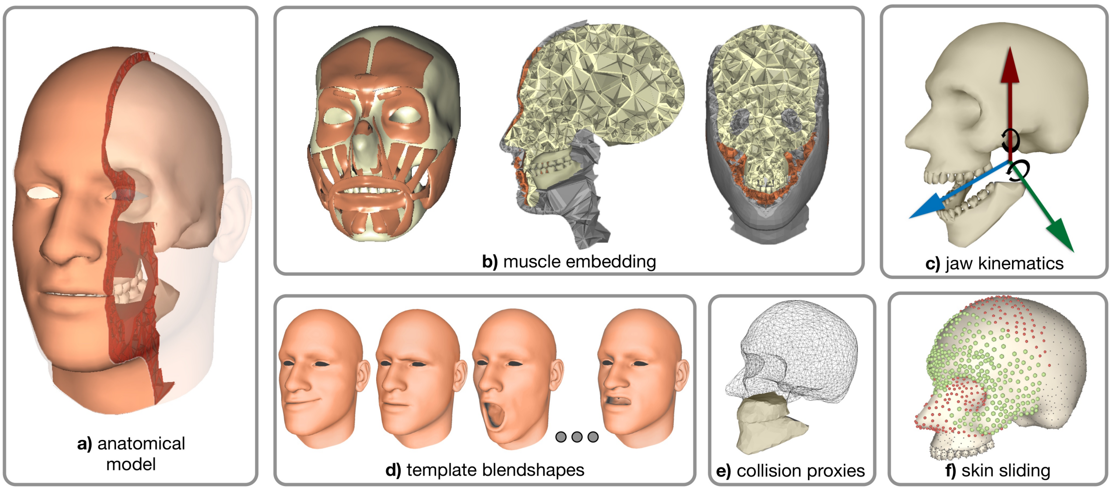


Figure 6.3 – Our template model consists of a volumetric representation of the tissue and bones (a), and a surface blendshape basis to represent the expression space (d). Muscles are embedded into a non-conforming tetrahedral mesh discretization (b). We explicitly model jaw kinematics with a 5 DoF joint (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).

comparisons to previous work. We show several application demos in Section 6.8, before concluding with a discussion of limitations and future work.

6.2 Related Work

Data-driven methods. A significant body of work in facial animation is based on data-driven techniques. Multi-view stereo acquisition systems are used extensively to acquire detailed geometry and texture models, e.g., [ABF*07, ARL*10, BBB*10]. Avatar creation based on simple cell-phone camera acquisition was proposed by Ichim et al. [IBP15], while depth sensors are often used to create 3D avatars suitable for realtime tracking, e.g., [WBLP11, BWP13, LYYB13]. These methods typically rely on data-driven priors to guide the reconstruction process, in particular morphable face models [BV99] or multi-linear (tensor) decomposition [VBPP05, CWZ*14]. They can be used in combination with upsampling methods, for example to add subject-specific details such as wrinkles [BBB*14]. Data-driven methods typically do not capture dynamic effects, even though some recent progress on this front has been made in the case of full-body animation [PMRMB15].

Advanced acquisition. In recent years we have witnessed significant improvements in acquisition of facial performance and morphology, in particular detailed skin microstructure [NFA*15], eyes [BBN*14, BBGB16], eyelids [BBK*15], hair [HMLL15], lips [GZW*16] and teeth [WBG*16b]. Modern methods can also capture medium-scale details (wrinkles) from monocular input in real-time [CBZB15]. Anatomical constraints have proven useful in estimating the rigid transformation of the skull (rigid stabilization) [BB14] and extracting detailed flesh deformations [WBGB16].

Anatomical models. Early anatomical models were based on procedural models such as FFD [CHP89]. Procedural muscle models were also used in pioneering work on facial reconstruction [KHS03]. Physics-based models of muscles and passive soft tissues were explored by Teran and colleagues [TBHF03, TSIF05, TSB*05], later extended into a comprehensive biomechanical model of the upper body [LST09], and combined with fluid simulation to study swimming [SLST14].

Biomechanical modeling is a complex task and several software platforms support soft tissue simulation, such as Sofa [ACF*07], ArtiSynth [LSF12], or FEBio [MEAW12]. An important aspect of soft tissue modeling is the capture of material properties [BBO*09] and their reproduction using modern fabrication methods such as 3D printing [BKS*12].

Algorithmic and numerical aspects of soft tissue simulation continue to be a topic of active research; recently, Fan et al. [FLP14] proposed an Eulerian-on-Lagrangian method to simulate dynamic musculoskeletal systems, while Saito et al. [SZK15] applied Projective Dynamics [BML*14] to simulate hypertrophy or atrophy of the muscles or fat.

More recently, Kadlecek et al. [KIL*16] studied the inverse problem of full-body modeling, inferring effects such as hypertrophy or atrophy of skeletal muscles from input 3D scans. Despite certain similarities to faces, a key difference is that full-body animation is characterized by muscles moving the bones, e.g., biceps moving the elbow. In facial animation, the skeletal articulation is limited to the jaw bone and facial expressions are created mainly by muscles pulling one another without any associated bone motion.

Physics-based face animation. The pioneering work of Sifakis et al. [SNF05] proposed a fully physics-based facial animation model, built from MRI and laser scan data of one specific subject. The key differences of our method are a more flexible muscle activation model combined with a more efficient inverse physics solver (Section 6.5). While Sifakis et al. [SNF05] also solve the inverse activation problem, their approach needs to invert a dense $n \times n$ matrix, where n is the number activation variables. This limits their method to using only low-dimensional activations, such as one activation per muscle, as opposed to our approach that allows for a richer high-dimensional activation model. A key benefit of our approach is that building the simulation model for different people is an almost entirely automatic process, as opposed to the substantial manual work required in previous work [SNF05]. Without the need of detailed parameter tuning, our approach also simplifies facial modifications such as slimming/fattening or geometric edits of the rigid bones.

The problem of scaling physics-based animation to different subjects has also been addressed in Cong et al. [CBB*15]. They propose a method that uses only the neutral expression to adapt an anatomical face model to different characters, including fictional ones. However, they do not attempt to closely match specific expressions of the target character as in our approach.

Cong and colleagues [CBF16] introduce “art-directed muscles”, i.e., blendshape models applied to the muscles. This approach caters to experienced visual artists who appreciate direct control over their anatomical rigs. However, the art-directed muscles lack translation and rotation invariance which limits their ability to generalize, e.g., to significant facial modifications or large external forces inducing displacements of entire muscle groups. We propose a translation and rotation invariant muscle activation model and an automatic inverse physics procedure for inferring muscle activations from target expressions.

Alternative approaches to physics-based facial animation use mass-spring systems [MWF*12] or finite element modeling of the face as elastic thin shell [BSC16]. While these methods support certain types of physics-based effects, a surface-only approach does not correctly handle collisions or support volumetric face modifications, such as visualizing the outcome of facial surgery. Modeling interior tissue and bones is also important when the face is subjected to inertial or external forces that visibly expose the rigidity of the internal bone structure.

Volumetric blendshapes as proposed in Ichim et al. [IKNDP16] introduce energy terms attracting deformation gradients to their target values derived from input facial expressions of a given person. The volumetric blendshapes are translation invariant, but they lack rotation invariance, introducing similar artifacts as linear elasticity, especially in situations with large external forces (Figure 6.10). In this paper, we create a model compatible with traditional blendshape interfaces, but we push the anatomical realism further by utilizing a novel muscle activation model, separating active and passive soft tissue layers, and introducing sliding constraints to attach soft tissue to the bones. As a consequence, our model implements a variety of advanced animation effects and supports significant modifications of the face simulation model, which enables a number of new applications as demonstrated in Section 6.8.

6.3 Template Face Model

Our approach starts from a *generic face model* – an anatomical face template corresponding to an average human subject (see Figure 6.3). We created this model from a commercially available anatomical data set [Zyg16] that contains polygonal representations of the bones (the skull, the jaw, including teeth), skin (including a realistic model of the oral cavity), and 33 facial muscles. Using the winding-number method of Jacobson et al. [JKSH13] 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 6.3-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.

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 6.3-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 $\mathbf{b} \in \mathbb{R}^5$.

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 [EF77] 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 6.4 and Section 6.5). This combination of surface blendshape basis and volumetric simulation model allows us retain compatibility with commonly used blendshape controls, while offering the benefits of advanced physics-based simulation effects.

6.4 Forward Physics

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 **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 **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

$$\begin{aligned} E_{\text{pass}}(\mathbf{x}) = \sum_i \min_{\mathbf{R}_i \in SO(3)} W_i^{\text{pass}} \mu \|\mathbf{F}_i(\mathbf{x}) - \mathbf{R}_i\|_F^2 + \\ W_i^{\text{pass}} \lambda (\det(\mathbf{F}_i(\mathbf{x})) - 1)^2, \end{aligned} \tag{6.1}$$

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. 6.1 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. $\mathbf{F}_i(\mathbf{x})$ denotes the deformation gradient, and $\mathbf{R}_i$ is an auxiliary rotation matrix used in the co-rotated model [SB12]. 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 6.8.

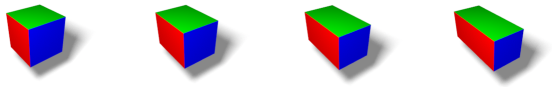
Active tissue. For tets corresponding to the active layer (muscles), we propose a novel activation model. Previous muscle models typically assume a given direction of muscle fibers along which the muscle contracts [TSB*05, LST09]. While this corresponds to the biological structure of muscles, the problem 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 [SZK15] 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 [SNF05]. 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).

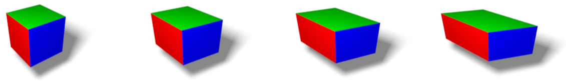
The property that the muscle cannot translate itself is already guaranteed by the translation invariance of deformation gradient operator $\mathbf{F}_i(\mathbf{x})$. Since a muscle tet should also not rotate itself, we require the activation to be a *symmetric* 3×3 matrix. Every symmetric matrix in $\mathbb{R}^{3\times 3}$ has an eigendecomposition of the form $\mathbf{Q}\tilde{}\mathbf{Q}^{\top}$, where $\mathbf{Q} \in SO(3)$ and $\tilde{} \in \mathbb{R}^{3\times 3}$ is diagonal. Therefore, the symmetric activation matrix corresponds to non-uniform scaling $(\tilde{})$ in an arbitrary orthonormal coordinate system ($\mathbf{Q}$). In other words, the symmetric matrix represents pure distortion without any change of orientation [SD92] (see Figure 6.4).

For each active tet, we define an activation vector $\mathbf{a}_i \in R^6$ and use a linear operator $\mathscr{S} : \mathbb{R}^6 \to \mathbb{R}^{3\times 3}$ to generate the corresponding symmetric matrix $\mathscr{S}(\mathbf{a}_i) \in \mathbb{R}^{3\times 3}$. Muscles, like most biological soft tissue, are approximately incompressible, which means that $\det(\mathscr{S}(\mathbf{a}_i)) = \det(\mathbf{Q}\tilde{}\mathbf{Q}^{\top}) = \det(\tilde{})$ should be close to 1. However, to compensate for discretization errors, we do not enforce $\det(\mathscr{S}(\mathbf{a}_i)) = 1$ strictly, but only as a soft constraint, as discussed in Section 6.5.

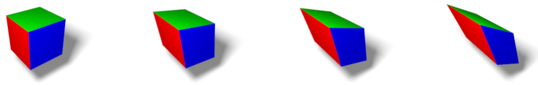
We use this activation model to define the deformation energy $E_{\text{act}}(\mathbf{x}, \mathbf{a})$ of active tissue, where $\mathbf{a}$ is a vector stacking the 6-dimensional activation parameters for all active tets. Specifi-



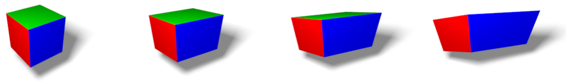
$\mathbf{a} = [2.0, 0.7, 0.7, 0.0, 0.0, 0.0]$



$\mathbf{a} = [1.7, 0.7, 1.0, 0.0, -0.6, 0.0]$



$\mathbf{a} = [1.8, 1.2, 0.8, 0.2, 0.1, 0.6]$



$\mathbf{a} = [0.9, 1.2, 1.7, 0.5, -0.5, 0.2]$

Figure 6.4 – Visualization of the capabilities of our 6-DoF activation model by squishing a cube, corresponding to a small sample of muscle tissue.

cally, we define:

$$\begin{aligned} E_{\text{act}}(\mathbf{x},\mathbf{a}) = \sum_i \min_{\mathbf{R}_i \in SO(3)} & W_i^{\text{act}} \mu \|\mathbf{F}_i(\mathbf{x}) - \mathbf{R}_i \mathscr{S}(\mathbf{a}_i)\|_F^2 + \\ & W_i^{\text{act}} \lambda (\det(\mathbf{F}_i(\mathbf{x})) - \det(\mathscr{S}(\mathbf{a}_i)))^2, \end{aligned} \tag{6.2}$$

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 $\mathbf{F}_i(\mathbf{x})$ with $\mathscr{S}(\mathbf{a}_i)$. The second term encourages the volume ratio of the deformed tet (i.e., $\det(\mathbf{F}_i(\mathbf{x}))$) to align with the volume ratio of the activation matrix $\det(\mathscr{S}(\mathbf{a}_i))$, 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 [SFCH13]. In animation, the visual importance of skin sliding is well recognized [LSNP13]. 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

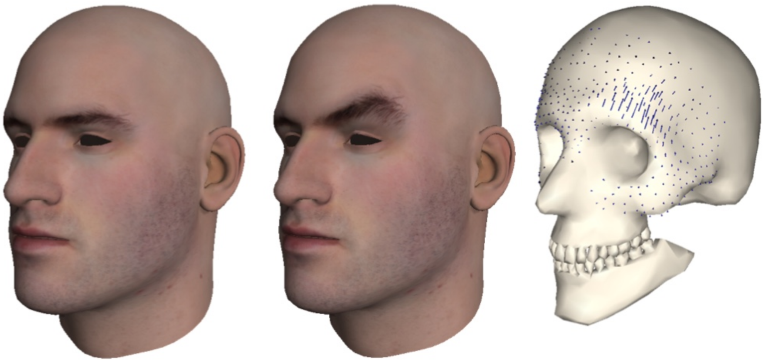


Figure 6.5 – An eyebrow raise expression uses the skin sliding feature of our model. The blue arrows show the displacement of the contact vertices between the cranium and the flesh.

amount of sliding displacement is generally small.

Formally, we express both pin and sliding constraints using a function $\mathbf{c}(\mathbf{x}, \mathbf{b})$ that depends also on the kinematic parameters $\mathbf{b} \in \mathbb{R}^5$ of the jaw bone. All of the constraints are satisfied if and only if $\mathbf{c}(\mathbf{x}, \mathbf{b}) = 0$. We have manually distributed the pin and sliding constraint as shown in Figure 6.3-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 6.5.

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 6.6. 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 6.5. Finding the steady state can be formulated as the following optimization problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & E_{\text{pass}}(\mathbf{x}) + E_{\text{act}}(\mathbf{x}, \mathbf{a}) + E_{\text{grav}}(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{c}(\mathbf{x}, \mathbf{b}) = 0, \mathbf{p}(\mathbf{x}) \geq 0, \end{aligned} \tag{6.3}$$

where $E_{\text{grav}}(\mathbf{x})$ represents a linear gravitational potential (i.e., the familiar mgh). The inequality constraints $\mathbf{p}(\mathbf{x})$ 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 $\mathbf{p}$. 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. 6.3 by alternating between an interior point solver used to minimize Eq. 6.3 for fixed collision constraints $\mathbf{p}$, and collision detection to update $\mathbf{p}$. We have initially implemented a "homebrew" augmented Lagrangian solver, but ultimately decided to use the IPOPT package [WB06], which has proven to be more robust and usually needs fewer iterations to converge.

Collisions - implementation details. We implemented collision handling between lips and the teeth and between the upper and lower lip, which are the areas most prone to interpenetration. Because the geometry of the teeth is quite detailed (and we are not aspiring to simulate e.g. flossing where the detail would be necessary), we start by creating proxy collision shapes for the upper and lower teeth (see Figure 6.3-e). The upper and lower lip geometries are already sufficiently smooth and we do not need any special collision proxy. We detect collisions using AABB hierarchies built for the upper and lower teeth proxy geometries and the upper and lower lips. Since lips are deforming, we recompute the AABB hierarchies at run-time. We did not use techniques to avoid or amortize this recomputation costs as this was not a bottleneck in our implementation.

For each of the collision proxies and the lips, we also manually define a "projection region", i.e., subset of triangles where interpenetrating vertices can be pushed to resolve collisions. Previous work considers all surface points as valid candidates for projection [MZS*11]. In our case this occasionally created problems such as resolving lip-teeth collisions by projecting the lip vertices behind the teeth, i.e., inside the mouth, which is rarely a plausible solution. Instead of more complicated continuous collision detection, we therefore disallowed these implausible projections. For each of the projection regions, we compute another AABB hierarchy that is used to find the closest point in the projection region, i.e., the location where an inter-penetrated vertex will be pushed in order to resolve the collision. A collision handling example during animation is shown in Figure 6.6.

6.5 Inverse Physics

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 $\mathbf{b}$ and muscle activations $\mathbf{a}$, which, when used in the forward simulation (Eq. 6.3), will produce a skin surface close to the input shape.

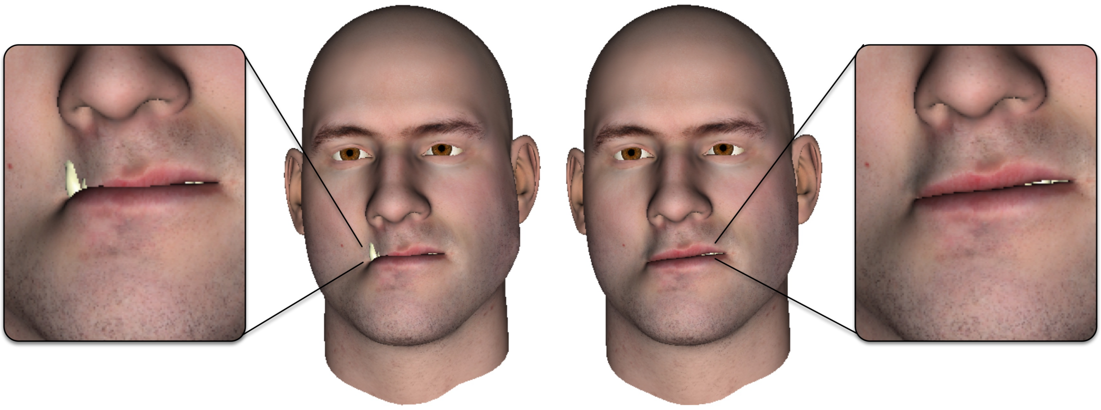


Figure 6.6 – Importance of collision handling. Without collisions, intersections between the teeth and the deformable tissue can occur (left). Our method correctly detects and handles the contact (right).

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

$$\begin{aligned} \min_{\mathbf{x},\mathbf{a},\mathbf{b}} \quad & \|\mathbf{S}\mathbf{x}-\mathbf{T}\mathbf{t}\|^2 + R_{\text{act}}(\mathbf{a}) + R_{\text{sparse}}(\mathbf{a}) \\ \text{subj. to} \quad & \mathbf{c}(\mathbf{x},\mathbf{b}) = 0, \mathbf{p}(\mathbf{x}) \geq 0 \\ & \nabla_{\mathbf{x}} E_{\text{pass}}(\mathbf{x}) + \nabla_{\mathbf{x}} E_{\text{act}}(\mathbf{x},\mathbf{a}) + \nabla_{\mathbf{x}} E_{\text{grav}}(\mathbf{x}) = 0 \end{aligned} \tag{6.4}$$

where $R_{\text{act}}(\mathbf{a})$ and $R_{\text{sparse}}(\mathbf{a})$ are regularization terms discussed below. The objective term $\|\mathbf{S}\mathbf{x}-\mathbf{T}\mathbf{t}\|^2$ measures how close state $\mathbf{x}$ is to the target $\mathbf{t}$. 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 [RL01]. 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 $\mathbf{t}$. The last vector equality constraint describes the condition of quasi-static equilibrium, i.e., the sum of all forces (gradients with respect to $\mathbf{x}$) is zero. Even though $\mathbf{x}$ is also an optimization variable, the desired output are the optimal values of muscle activations $\mathbf{a}$ and bone parameters $\mathbf{b}$.

Regularization. Without regularization, the optimization of Eq. 6.4 can lead to over-fitting and anatomically implausible activations $\mathbf{a}$. To provide an appropriate prior on activation patterns, we exploit the geometric structure of the muscles by estimating an approximate preferred contraction direction. Following Choi et al. [CB13] we compute these directions by solving a Laplace equation and encode the corresponding orientations for the i-th tet as

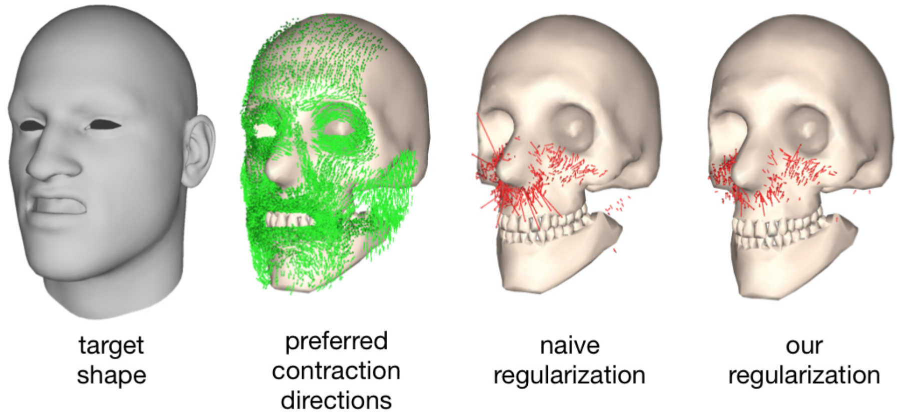


Figure 6.7 – Muscle activation regularization. Red lines indicate the direction and magnitude of the dominant muscle contraction, computed from the SVD of the activation matrix.

$\mathbf{Q}_i \in SO(3)$. The regularizing prior softly penalizes deviations in muscle contraction from the preferred direction and is defined as:

$$R_{\text{act}}(\mathbf{a}) = \sum_{i,m} f_{i,m} \left\| \mathbf{Q}_i^\top \begin{bmatrix} \gamma_{m(i)}^{-1} & 0 & 0 \\ 0 & \sqrt{\gamma_{m(i)}} & 0 \\ 0 & 0 & \sqrt{\gamma_{m(i)}} \end{bmatrix} \mathbf{Q}_i - \mathscr{S}(\mathbf{a}_i) \right\|^2$$

where i sums over all active tets and m over all muscles. Since our tet mesh does not conform to the muscles, some tets may be occupied only partially by a muscle, or be occupied by several muscles. We calculate the fraction $f_{i,m} \in [0,1]$ of tet i occupied by muscle m by Monte-Carlo sampling (if an active tet contains some amount of passive tissue, we get $\sum_m f_{i,m} < 1$ and the regularization strength is proportionally reduced as expected). The contraction parameters $\gamma_m \geq 0$ are auxiliary variables representing the contraction of muscle m. Recall that $\mathscr{S}(\mathbf{a})$ is our symmetric muscle activation matrix introduced in Section 6.4. Intuitively, $R_{\text{act}}(\mathbf{a})$ encourages all tets corresponding to a single muscle to contract in a uniform, volume preserving way (because $\gamma_{m(i)}^{-1}\sqrt{\gamma_{m(i)}}\sqrt{\gamma_{m(i)}} = 1$).

In addition to the muscle-activation regularization term R_{act}, we found it beneficial to also include the following term to promote sparse muscle activations:

$$R_{\text{sparse}}(\mathbf{a}) = \sum_m \sqrt{\sum_i f_{i,m} \|\mathbf{a}_i\|^2} \tag{6.5}$$

Specifically, this is a group sparsity term similar to L_1 regularization, but applied to entire groups – in our case, muscles. This term encourages all activations corresponding to one muscle to remain zero unless contributing significantly to the result. We introduced this term to avoid small spurious activations of remote muscles, which is justified when our target

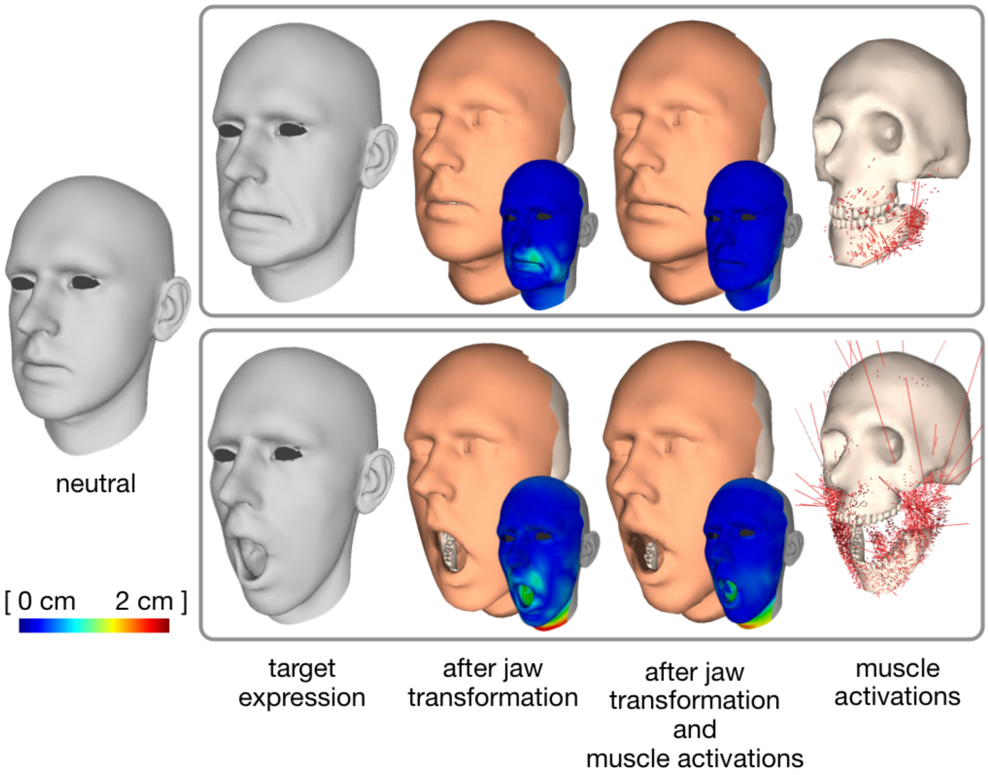


Figure 6.8 – Inverse physics finds jaw transformation and muscle activations that accurately reproduce the target blendshapes.

shapes **t** correspond to traditional FACS-type blendshape models which isolate individual action units. Figure 6.7 shows that compared to a naive L_2 regularization approach, our method leads to sparser activations that are better aligned with the geometric structure of the muscles.

Numerical solution. As in Section 6.4, we use interior-point methods [WB06] to solve the constrained optimization problem in Eq. 6.4. Our implementation of the Hessian of the Lagrangian of Eq. 6.4 ignores third-order derivatives of E (pretends they are zero), amounting to the commonly used Gauss-Newton approximation of the Hessian [SNF05, BKS*12]. We alternate the interior point solver with collision detection that determines the non-penetration constraints **p** as in Section 6.4.

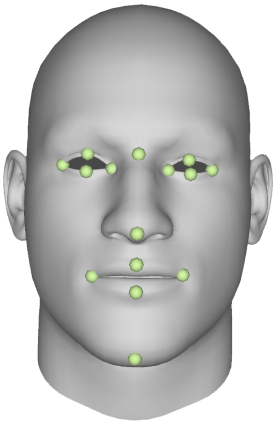
Even though including the regularization term R_{act} could be directly incorporated into our optimization objective (adding γ_m as auxiliary variables), we found that this significantly increases the non-linearity of the problem and forces the non-linear solver to take many iterations, each making only slow progress towards the solution. To avoid this problem, we instead use a local-global approach [SA07]. In the local step, the activations $\mathbf{a}$ are fixed and we compute optimal γ_m by finding roots of a 6-th order polynomial using the method of Brent [Bre71]. In the global step, we call the interior point solver to optimize $\mathbf{a}$ for fixed γ_m, which is an easier optimization problem exhibiting fast convergence.

Figure 6.8 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.

6.6 Phace 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.

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. [IKNDP16]. Each of the scans is approximately aligned with the skin of our template model (Section 6.3) using rigid registration (plus uniform scale). Then we apply non-rigid ICP [RL01] 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 $\mathbf{s}_{\text{neut}}$ for the neutral and $\mathbf{s}_k$ for k-th expression.



Next, we deform our volumetric template model such that its boundary (skin) aligns with $\mathbf{s}_{\text{neut}}$. This is accomplished with Anatomy Transfer [DLG*13, IKNDP16]. 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 6.4). We then use Example-Based Facial Rigging [LWP10] to convert the registered expressions $\mathbf{s}_k$ to subject-specific blendshapes $\mathbf{c}_j$, $j = 1, \ldots, 48$.

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 $\mathbf{a}_j$ and jaw bone parameters $\mathbf{b}_j$ that correspond to each of the blendshapes $\mathbf{c}_j$ using the Inverse Physics optimization of Section 6.5.

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 6.4 for each animation frame. Given per-frame blendshape weights $\mathbf{w} = \{w_1, \ldots, w_{48}\}$, we compute muscle activations as $\mathbf{a} = \mathbf{a}_{\text{neut}} + \sum_j w_j(\mathbf{a}_j - \mathbf{a}_{\text{neut}})$, where $\mathbf{a}_{\text{neut}}$ corresponds to neutral activations, i.e., each activation $\mathscr{S}(\mathbf{a}_{j,i}) = \mathbf{I} \in \mathbb{R}^{3\times 3}$. Linear blending of the activation parameters is justified because there is no rotational component in symmetric matrices [SD92]. Similarly, we compute the blended jaw kinematics parameters $\mathbf{b} = \sum_j w_j \mathbf{b}_j$. 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. 6.3. We use the popular backward Euler integration, which in its optimization form [LBOK13] corresponds to augmenting the objective of Eq. 6.3 with the term: $\frac{1}{2}\|\mathbf{x} - (\mathbf{x}_n + h\mathbf{v}_n)\|^2_{\mathbf{M}}$, where $\mathbf{x}_n$ and $\mathbf{v}_n$ 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 $\mathbf{x}$ of Eq. 6.3 then becomes the new state $\mathbf{x}_{n+1}$ and the new velocity is $\mathbf{v}_{n+1} = (\mathbf{x}_{n+1} - \mathbf{x}_n)/h$. The main difference from the quasi-static solution is that the dynamic solution depends on the previous state $(\mathbf{x}_n, \mathbf{v}_n)$, 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 [BML*14] and change the additional term to $\frac{1}{2}\|\mathbf{x} - (\mathbf{x}_n + h\mathbf{v}_n + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}})\|^2_{\mathbf{M}}$. Here $\mathbf{f}_{\text{ext}} \in \mathbb{R}^3$ 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 $\mathbf{F}_{\text{total}}(\mathbf{x})$ is assumed to be composed of an elastic deformation component and plastic deformation component, i.e., $\mathbf{F}_{\text{total}}(\mathbf{x}) = \mathbf{F}_{\text{elast}}(\mathbf{x})\mathbf{F}_{\text{plast}}$ or, equivalently, $\mathbf{F}_{\text{elast}}(\mathbf{x}) = \mathbf{F}_{\text{total}}(\mathbf{x})\mathbf{F}^{-1}_{\text{plast}}$. Note that $\mathbf{F}_{\text{plast}}$ does not depend on the current deformed state $\mathbf{x}$. The deformation gradient $\mathbf{F}_i(\mathbf{x})$ used in Eq. 6.1 and Eq. 6.2 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 $\mathbf{F}_i(\mathbf{x})$ in Eq. 6.1 and Eq. 6.2 by $\mathbf{F}_i(\mathbf{x})\mathbf{F}^{-1}_{\text{plast},i}$, 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 6.8. Plasticity, as well as inertia and external forces are applied in forward physics only.

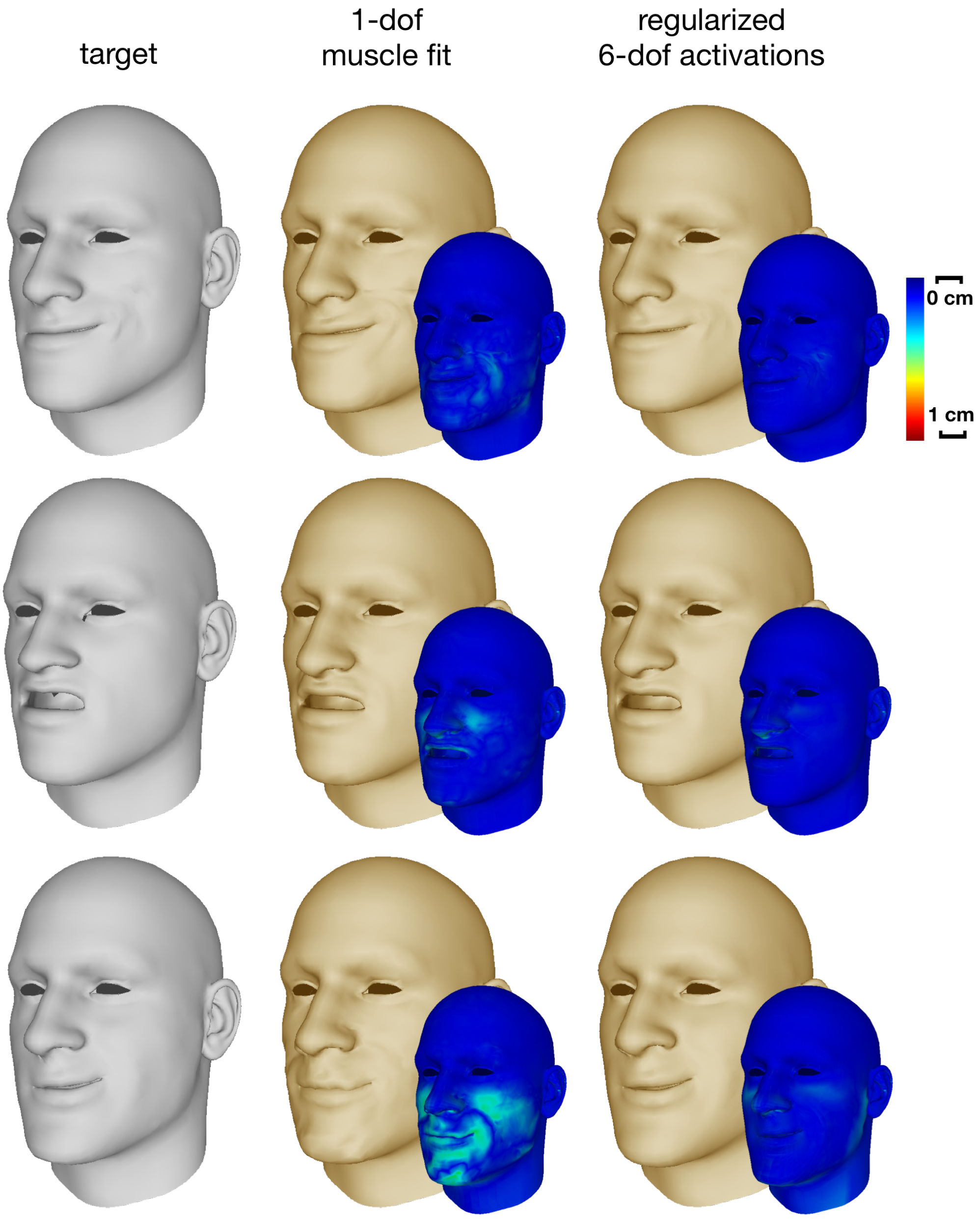


Figure 6.9 – Our 6-DoF muscle activation model (middle) leads to more accurate reconstruction of the target expression (right) than previous 1-DoF fiber-aligned activations models (left).

6.7 Evaluation

Before showing application results of our method in Section 6.8, we evaluate the behavior of our optimization algorithms and provide comparisons to previous work.

Muscle activation model. As mentioned in Section 6.4, previous methods constrain the deformation along muscle fibre directions [TSB*05, SNF05, LST09, SZK15]. 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 6.9 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 [CB13] (these directions are shown in Figure 6.7). As Figure 6.9 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. [IKNDP16] 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 6.10. 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 [IKNDP16] assumes that all soft tissue can activate. In addition, our approach includes a kinematic model for the jaw, whereas Ichim et al. [IKNDP16] 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 6.5).

Model adaptations. Our approach supports animating a character after significant modifications of the neutral pose (e.g. slimming/fattening, bone modifications, see Section 6.8) using the same muscle activation patterns. One might argue that the same effects could be obtained by using deformation transfer [SP04] on traditional linear animation models. For example, similar 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 6.11.

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

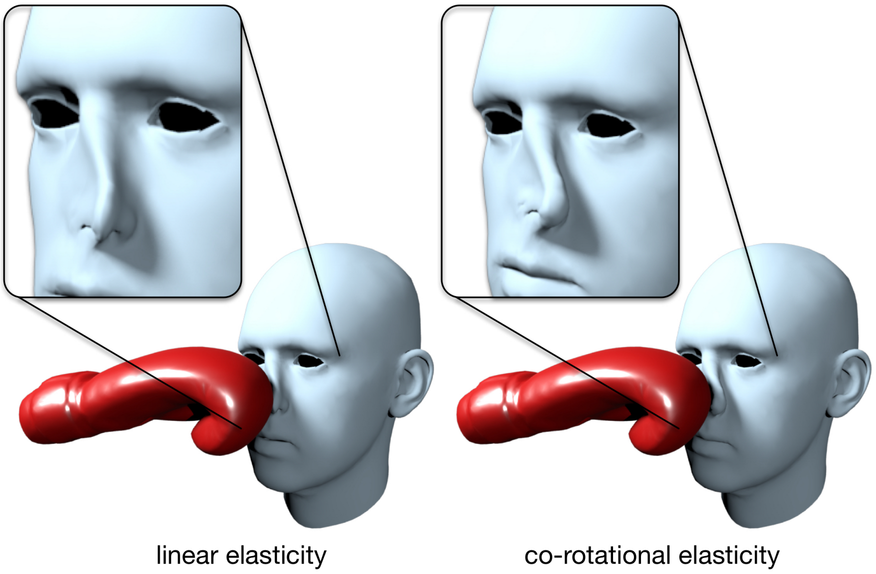


Figure 6.10 – A boxing punch to the nose results in artifacts with an elastic model lacking rotation invariance as in [IKNDP16] (left). More realistic deformations are obtained with our rotation-invariant model (right).

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 paper has 8098 vertices and 35626 tetrahedra. The active muscle layer covers approx. 27% of the entire flesh. The surface mesh model of the entire skin has 6393 vertices and 12644 faces.

6.8 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

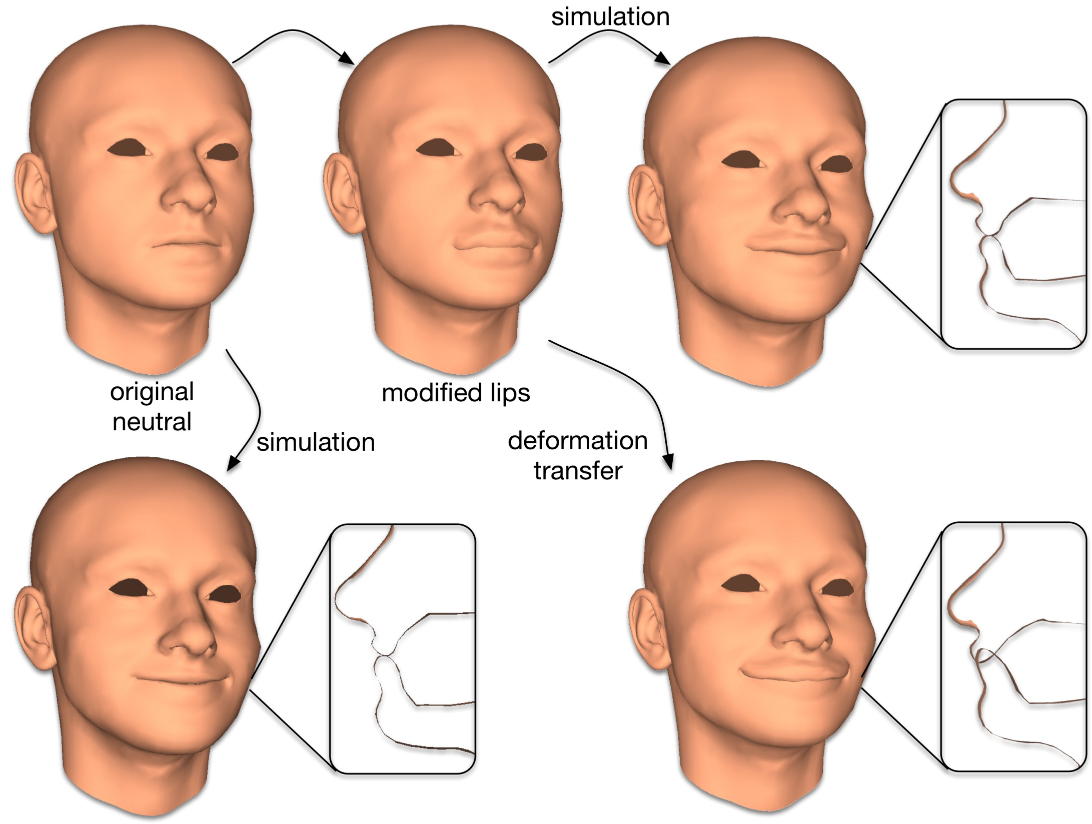


Figure 6.11 – Model adaptations such as increased lip volume are handled accurately in our approach, while deformation transfer [SP04] leads to self-intersections.

be difficult to obtain with purely generative geometric methods. Please also refer to the accompanying video 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. [WBLP11]. 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 6.12-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

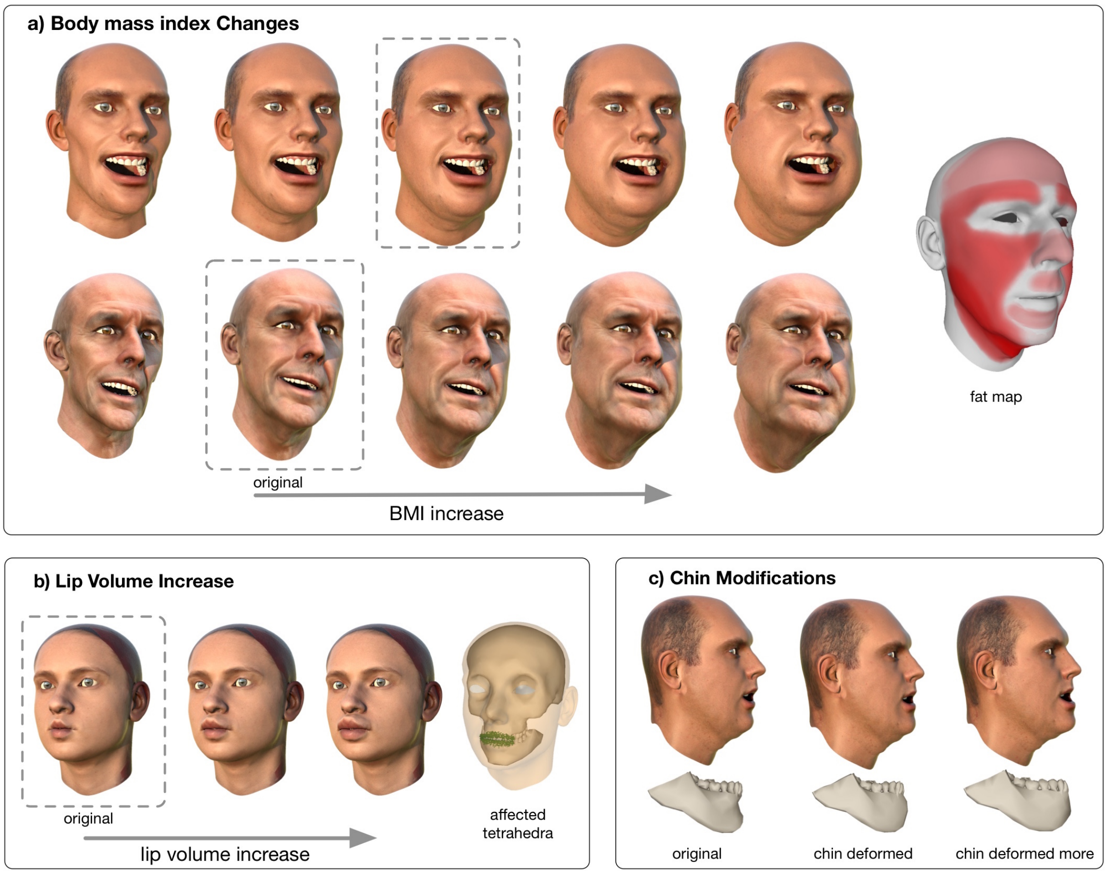


Figure 6.12 – Application Demos I: a) Body mass index changes and their impact on expressions. The original avatar is highlighted with dashed lines. More intense red in the fat map means more volume change of the corresponding face region. b) Results of lip injection, where affected tets are shown in green on the right. c) Effects of modifying the rigid bone structure of the chin.

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 tet-mesh by a diffusion process, similar to standard polygon-mesh diffusion flow ([BKP*10], 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 6.12-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

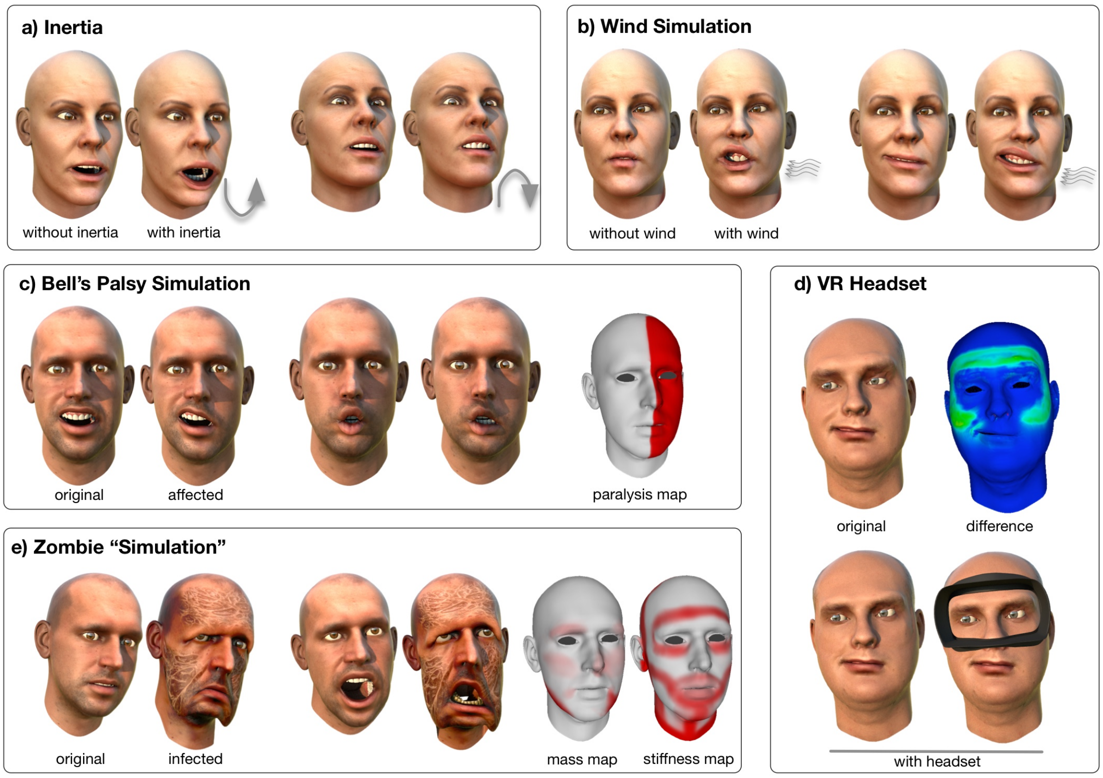


Figure 6.13 – a) Simulating inertia under sudden motion changes (e.g., jumping). b) Dynamic deformations in a wind force field. c) Simulating Bell's Palsy affecting half of the face of an actor. d) VR headset obstructing the full motion of expressions on the face. e) Artistic editing to create a zombie character by adapting the mass and stiffness distribution as indicated in the color-coded maps.

Figure 6.12-a. For slimming, we keep the default stiffness $\mu = 1$.

Facial surgery. Figures 6.12-b and 6.12-c demonstrate potential applications in visualizing the possible outcomes of facial surgery, helping patients to choose between different versions of corrective or cosmetic procedures. Our method allows direct manipulation of the deformable soft tissue (e.g. lip fat injection) or the rigid bones (e.g. chin displacement). The simulation then provides a detailed visual preview of such interventions on the expression dynamics of the animated person. We modeled the lip fat injection using our plasticity model and diffusion tool, simulating the process of injecting filler material with a syringe through several points on the skin. To simulate the lower stiffness of the fat-like filler material, we decreased the soft tissue stiffness μ to 0.8 for the medium, and to 0.5 for the high lip volume effect (Figure 6.12-b). For the chin displacement we directly edited the bone using an interactive mesh modeling tool.

Inertia. Figure 6.13-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 6.13-b shows how an animation can be augmented with complex external force interactions produced by a dynamic wind field. Figure 6.13-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 6.13-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 6.13-e. We designed two texture maps to modulate the mass and stiffness (see Figure 6.13-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 $1 - 3 g/cm^3$, achieving artistic “undead” effects.

6.9 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. A potentially more practical 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 [BV99].

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, and generalizing our model to full body simulations.

6.10 Conclusion

We propose 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.

We advocate the use of physics-based simulation as a principled solution to these issues. Our experiments validate that this approach leads to high-quality facial animations and facilitates new editing capabilities, e.g., by manipulating the face model's physical structure or its dynamic behavior. These advanced effects come at the cost of increased computational overhead. However, as computational power increases and algorithms are improved, we believe that the simulation-based approach to facial animation will become more and more viable in the future. This path certainly offers a rich set of opportunities for future research with applications not only in movies and games, but also in surgery simulation, interactive therapy, sports, or biomedical research.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-1617172 and IIS-1622360, the grant SVV-2017-260452 and GA UK 1524217. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also gratefully acknowledge the support of Activision.

Retrospective

This latest publication of my PhD can be seen as the culmination of my recent work on physics-based facial animation. It combines numerous concepts we have developed in all the previous projects, as it will be succintly explained in the next paragraphs.

First of all, in terms of input data, this last project showed results by using data collected in previous projects. We have used blendshape models created from smartphone images and videos, as presented in Chapter 2 [IBP15]. For higher quality data we turned to our homebrew stereo scanner from Ichim et al. [IKNDP16]. The registration technique was also the one based on Projective Dynamics FEM, as in Chapter 5.

The inverse physics mathematical formulation was inspired by and improved upon our previous work on anatomical bodies from Chapter 4 [KIL*16].

7 Conclusions

7.1 Summary

The work that has been presented in this thesis can be split into two themes:

- *human face and body avatar creation using lightweight acquisition approaches.* This part focused on the computer vision problems around extracting as much information as possible from noisy low-quality optical data.
- *physics-based anatomically-inspired animation models for digital human avatar reconstruction and animation.* This second part introduced novel physics optimization techniques in order to represent the avatars as simulation-ready objects, and then obtain compelling animations through a physics optimization process.

In Chapter 2 we have looked into a complete pipeline for reconstructing fully rigged, personalized 3D facial avatars using image and video data coming from an off-the-shelf smartphone. We have shown how a blendshape template can be adapted to the video recording through an optimization that includes multiple types of visual cues. Furthermore, we have integrated wrinkles as UV detail maps using a regression technique.

Chapter 3 presented a complete framework for tracking and modeling articulated human bodies from RGB-D video data. We have investigated how L1 regularization can help generate a sparse description of the body shape by using a blendshape body model.

The first approach corresponding to physics-based animation was explained in Chapter 5. We argued that linear animation models have a lot of limitations when the application requires features such as dynamics, collision response, or incompressibility of the flesh. For this reason, we propose a volumetric head template model and show how this can be used to register facial scans and produce realistic animations, while automatically fixing issues such as the ones coming from volume loss or collisions.

Turning to full-body avatars, in Chapter 4 we presented a method for creating personalized anatomical models that can be animated through physics simulation. We have explained our key contribution, which is formulating and solving a large-scale optimization problem that we dubbed *inverse physics*.

Finally, Chapter 6 extended the mathematical concepts from inverse physics for usage with facial animation. In this project, the digital heads produce expressions actuated by active volumetric elements as well as bone kinematics. We showed how we can incorporate complex physical effects such as non-linear interactions between bones, passive fat and active muscles, as well as skin sliding and collisions. Lastly, we showed how easy it is to generate complex animations by artistically altering elements in the physics-domain, as opposed to traditional keyframing and mesh editing.

7.2 Future Work

We believe this thesis opens up a lot of interesting possibilities for future work directions, and here we make some suggestions.

Regarding lightweight acquisition, the presented work relied only on optical sensor information for the reconstruction and tracking of the user. By using the other sensors commonly available on smartphones (e.g., compass, accelerometer) should help improve the reconstruction performance and quality. Moreover, by making use of inertial data, the multiview stereo problem posed for the neutral reconstruction could be initialized with a good guess of the camera motion and the performance is expected to improve dramatically.

An immediate criticism to the physics-based animation work is the lack of a quantitative evaluation. In the corresponding publications it has been shown that the proposed physics-based approaches are able to generate all the animations that their linear counterparts were able to. In addition, they bring the big advantage that the new models can perform very well under extrapolation with external forces. Those extrapolations have only been evaluated qualitatively in this work, as the main target application was in the entertainment sector (i.e., video games, movie special effects). We believe exploring medical directions is of great interest. For this we will need a better evaluation with a tight feedback loop with real-world measurements. E.g., measure muscle fiber directions for a specific person, muscle contraction shapes under different expressions using MRI volumetric scans, muscle contraction patterns with electrodes etc.. Along the same lines, an important possible venue is to build a complete avatar of a single subject by using recent volumetric measurement techniques. It would partially involve re-creating the seminal work of Teran and Sifakis, but with access to modern scanning techniques, better understanding of numerical optimization, as well as more computing power.

Recently, it has been proven that deep learning is able to solve multiple of the longstanding machine learning and computer vision problems. Recent literature proposes machine learning approaches for improving the performance of fluid or general physics simulation [TSSP16, JSP*15]. Moreover, [WBGB16] suggest a method to encode subspaces for patches of face skin to be used for mononcular tracking. We believe that our physics-based animation models can be sped up or used as training for learning-based techniques, which is essential for realtime applications.

Another possible research direction would be to better simulate the interactions between clothes and bodies. Usually bodies are used as collision proxies to compute the sliding constraints for the clothes to drape on them [KKN*13, BMF03], but the bodies never respond to interactions with the clothes. Now that we have proposed good approximations of the different elastic components of human bodies, an immediate application would be to look into simulating tight-fitting and body-shaping clothes such as sportswear.

7.3 Final Remarks

To conclude, this thesis proposes an approach to the lightweight digital human reconstruction and animation problem, as well as methods for compelling physics-based character animation. We believe that simulation-based systems like ours will become more viable in the future as hardware performance increases and the algorithms become more efficient. This path certainly offers a rich set of opportunities for future research with applications not only in movies and games, but also in surgery simulation, interactive therapy, sports, or biomedical research.

Bibliography

[ABF*07] AMBERG B., BLAKE A., FITZGIBBON A., ROMDHANI S., VETTER T.: Reconstructing high quality face-surfaces using model based stereo. In *International Conference on Computer Vision* (2007), IEEE, pp. 1–8.

[ACF*07] ALLARD J., COTIN S., FAURE F., BENSOUSSAN P.-J., POYER F., DURIEZ C., DELINGETTE H., GRISONI L.: Sofa-an open source framework for medical simulation. In *MMVR 15-Medicine Meets Virtual Reality* (2007), vol. 125, IOP Press, pp. 13–18.

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 22*, 3 (2003), 587–594.

[AFB*13] ALEXANDER O., FYFFE G., BUSCH J., YU X., ICHIKARI R., JONES A., DEBEVEC P., JIMENEZ J., DANVOYE E., ANTIONAZZI B., EHELER M., KYSELA Z., VON DER PAHLEN J.: Digital ira: Creating a real-time photoreal digital actor. In *ACM SIGGRAPH 2013 Posters* (2013).

[ARL*09] ALEXANDER O., ROGERS M., LAMBETH W., CHIANG M., DEBEVEC P.: Creating a photoreal digital actor: The digital emily project. In *Visual Media Production, 2009. CVMP'09. Conference for* (2009).

[ARL*10] ALEXANDER O., ROGERS M., LAMBETH W., CHIANG J.-Y., MA W.-C., WANG C.-C., DEBEVEC P.: The digital Emily project: Achieving a photorealistic digital actor. *Computer Graphics and Applications, IEEE 30*, 4 (2010), 20–31.

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: shape completion and animation of people. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2005), vol. 24, pp. 408–416.

[BB14] BEELER T., BRADLEY D.: Rigid stabilization of facial expressions. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 33*, 4 (2014), 44.

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 29*, 4 (2010), 40.

[BBB*14] BERMANO A. H., BRADLEY D., BEELER T., ZUND F., NOWROUZEZAHRAI D., BARAN I., SORKINE-HORNUNG O., PFISTER H., SUMNER R. W., BICKEL B., ET AL.: Facial performance enhancement using dynamic shape space analysis. *ACM Transactions on Graphics 33*, 2 (2014), 13.

[BBGB16] BÉRARD P., BRADLEY D., GROSS M., BEELER T.: Lightweight eye capture using a parametric model. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 35*, 4 (2016), 117.

[BBK*15] BERMANO A., BEELER T., KOZLOV Y., BRADLEY D., BICKEL B., GROSS M.: Detailed spatio-temporal reconstruction of eyelids. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015), 44.

[BBN*12] BEELER T., BICKEL B., NORIS G., BEARDSLEY P., MARSCHNER S., SUMNER R. W., GROSS M.: Coupled 3d reconstruction of sparse facial hair and skin. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2012).

[BBN*14] BÉRARD P., BRADLEY D., NITTI M., BEELER T., GROSS M.: High-quality capture of eyes. *Proceedings of ACM SIGGRAPH Asia 33*, 6 (Nov. 2014), 223:1–223:12.

[BBO*09] BICKEL B., BÄCHER M., OTADUY M. A., MATUSIK W., PFISTER H., GROSS M.: Capture and modeling of non-linear heterogeneous soft tissue. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 28*, 3 (2009).

[BF14] BIEHLER J., FANE B.: *3D Printing with Autodesk: Create and Print 3D Objects with 123D, AutoCAD and Inventor*, 1st ed. Que Publishing Company, 2014.

[BHB*11] BEELER T., HAHN F., BRADLEY D., BICKEL B., BEARDSLEY P., GOTSMAN C., SUMNER R. W., GROSS M.: High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2011).

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon mesh processing*. CRC press, 2010.

[BKS*12] BICKEL B., KAUFMANN P., SKOURAS M., THOMASZEWSKI B., BRADLEY D., THE T., JACKSON P., MARSCHNER S., MATUSIK W., GROSS M.: Physical face cloning. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 31*, 4 (2012), 118.

[BLB*08] BICKEL B., LANG M., BOTSCH M., OTADUY M. A., GROSS M. H.: Pose-space animation and transfer of facial details. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2008).

[BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2003), Eurographics Association, pp. 28–36.

[BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 33*, 4 (2014), 154.

[Bre71] BRENT R. P.: An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal 14*, 4 (1971).

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition* (2014).

[BRM08] BASTIONI M., RE S., MISRA S.: Ideas and methods for modeling 3d human figures: the principal algorithms used by makehuman and their implementation in a new approach to parametric modeling. In *Proceedings of the 1st Bangalore Annual Compute Conference* (2008), ACM, p. 10.

[BSC16] BARRIELLE V., STOIBER N., CAGNIART C.: Blendforces, a dynamic framework for facial animation. *Computer Graphics Forum* (2016).

[BTP14] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Dynamic 2d/3d registration. *Eurographics Tutorial* (2014).

[Bun05] BUNNELL M.: Dynamic ambient occlusion and indirect lighting. *Gpu gems* (2005).

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 187–194.

[BWP13] BOUAZIZ S., WANG Y., PAULY M.: Online modeling for realtime facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 32*, 4 (2013), 40.

[BY86] BRUCE V., YOUNG A.: Understanding face recognition. *British journal of psychology 77*, 3 (1986), 305–327.

[CB13] CHOI H. F., BLEMKER S. S.: Skeletal muscle fascicle arrangements can be reconstructed using a laplacian vector field simulation. *PloS one 8*, 10 (2013), e77576.

[CBB*15] CONG M., BAO M., BHAT K. S., FEDKIW R., ET AL.: Fully automatic generation of anatomical face simulation models. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2015), ACM, pp. 175–183.

[CBF16] CONG M., BHAT K. S., FEDKIW R.: Art-directed muscle simulation for high-end facial animation. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2016), pp. 119–127.

[CBZB15] CAO C., BRADLEY D., ZHOU K., BEELER T.: Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015), 46.

[CCC*10] CHAMBOLLE A., CASELLES V., CREMERS D., NOVAGA M., POCK T.: An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery 9* (2010), 263–340.

[CDHR08] CHEN Y., DAVIS T. A., HAGER W. W., RAJAMANICKAM S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS) 35*, 3 (2008), 22.

[CHP89] CHADWICK J. E., HAUMANN D. R., PARENT R. E.: Layered construction for deformable animated characters. In *Computer Graphics (Proceedings SIGGRAPH)* (1989), vol. 23, ACM, pp. 243–252.

[CHZ14] CAO C., HOU Q., ZHOU K.: Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2014).

[CPSS10] CHAO I., PINKALL U., SANAN P., SCHRÖDER P.: A simple geometric model for elastic deformations. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2010), vol. 29, ACM, p. 38.

[CWLZ13] CAO C., WENG Y., LIN S., ZHOU K.: 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2013).

[CWWS12] CAO X., WEI Y., WEN F., SUN J.: Face alignment by explicit shape regression. In *Computer Vision and Pattern Recognition* (2012).

[CWZ*14] Cao C., Weng Y., Zhou S., Tong Y., Zhou K.: Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* (2014).

[CY08] Chartrand R., Yin W.: Iteratively reweighted algorithms for compressive sensing. In *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on* (2008), IEEE, pp. 3869–3872.

[CZXZ14] Chen X., Zheng C., Xu W., Zhou K.: An asymptotic numerical method for inverse elastic shape design. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 33*, 4 (2014), 95.

[CZZ14] Chai M., Zheng C., Zhou K.: A reduced model for interactive hairs. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (July 2014).

[DAA*07] Delp S. L., Anderson F. C., Arnold A. S., Loan P., Habib A., John C. T., Guendelman E., Thelen D. G.: Opensim: open-source software to create and analyze dynamic simulations of movement. *Biomedical Engineering, IEEE Transactions on 54*, 11 (2007), 1940–1950.

[DDB*15] Deuss M., Deleuran A. H., Bouaziz S., Deng B., Piker D., Pauly M.: Shapeop—a robust and extensible geometric modelling paradigm. In *Modelling Behaviour*. Springer, 2015, pp. 505–515.

[DGCV*06] De Greef S., Claes P., Vandermeulen D., Mollemans W., Suetens P., Willems G.: Large-scale in-vivo caucasian facial soft tissue thickness database for craniofacial reconstruction. *Forensic science international 159* (2006), S126–S146.

[DH72] Duda R. O., Hart P. E.: Use of the hough transformation to detect lines and curves in pictures. *Communications of ACM* (1972).

[DIF04] Dimitrijevic M., Ilic S., Fua P.: Accurate face models from uncalibrated and ill-lit video sequences. In *Computer Vision and Pattern Recognition* (2004), vol. 2, IEEE, pp. II–II.

[DLG*13] Dicko A.-H., Liu T., Gilles B., Kavan L., Faure F., Palombi O., Cani M.-P.: Anatomy transfer. *Proceedings of ACM SIGGRAPH Asia 32*, 6 (2013), 188.

[DOKA13] Douvantzis P., Oikonomidis I., Kyriazis N., Argyros A.: Dimensionality reduction for efficient single frame hand pose estimation. In *Computer Vision Systems*. Springer, 2013, pp. 143–152.

[EF77] Ekman P., Friesen W. V.: Facial action coding system.

[FLP14] Fan Y., Litven J., Pai D. K.: Active volumetric musculoskeletal systems. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 33*, 4 (2014), 152.

[FP10] Furukawa Y., Ponce J.: Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010).

[FSB04] Frolova D., Simakov D., Basri R.: Accuracy of spherical harmonic approximations for images of lambertian objects under far and near lighting. In *European Conference on Computer Vision*. 2004.

[Fu98] Fu W. J.: Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics 7*, 3 (1998), 397–416.

[Fun13] Fung Y.-c.: *Biomechanics: mechanical properties of living tissues.* Springer Science & Business Media, 2013.

[GFT*11] Ghosh A., Fyffe G., Tunwattanapong B., Busch J., Yu X., Debevec P.: Multiview face capture using polarized spherical gradient illumination. In *Proceedings of ACM SIGGRAPH Asia* (2011).

[GJ*10] Guennebaud G., Jacob B., et al.: Eigen v3. http://eigen.tuxfamily.org, 2010.

[Goo17] Google: Tango, 2017. [Online; accessed 01-Jul-2017].

[Gra06] Gray R. M.: *Toeplitz and circulant matrices: A review.* now publishers Inc, 2006.

[GVWT13] Garrido P., Valgaerts L., Wu C., Theobalt C.: Reconstructing detailed dynamic face geometry from monocular video. *Proceedings of ACM SIGGRAPH Asia 32*, 6 (2013), 158.

[GW06] Gonzalez R. C., Woods R. E.: *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., 2006.

[GWBB09] Guan P., Weiss A., Balan A. O., Black M. J.: Estimating human shape and pose from a single image. In *International Conference on Computer Vision* (2009), IEEE, pp. 1381–1388.

[GZW*16] Garrido P., Zollhöfer M., Wu C., Bradley D., Pérez P., Beeler T., Theobalt C.: Corrective 3d reconstruction of lips from monocular video. *Proceedings of ACM SIGGRAPH Asia 35*, 6 (2016), 219.

[HBB*13] HELTEN T., BAAK A., BHARAJ G., MULLER M., SEIDEL H., THEOBALT C.: Personalization and evaluation of a real-time depth-based full body tracker. In *Proceedings of the International Conference on 3D Vision (3DV)* (2013).

[HBLB17] HU L., BRADLEY D., LI H., BEELER T.: Simulation-ready hair capture. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 281–294.

[HCTW11] HUANG H., CHAI J., TONG X., WU H.-T.: Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2011).

[HLRB12] HIRSHBERG D. A., LOPER M., RACHLIN E., BLACK M. J.: Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *European Conference on Computer Vision*. Springer, 2012, pp. 242–255.

[HMLL14] HU L., MA C., LUO L., LI H.: Robust hair capture using simulated examples. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2014).

[HMLL15] HU L., MA C., LUO L., LI H.: Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015).

[HRD*12] HOLZER S., RUSU R. B., DIXON M., GEDIKLI S., NAVAB N.: Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (2012), IEEE, pp. 2684–2689.

[HSS*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 337–346.

[IBP15] ICHIM A. E., BOUAZIZ T., PAULY M.: Dynamic 3d avatar creation from handheld video input. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2015).

[IKNDP16] ICHIM A., KAVAN L., NIMIER-DAVID M., PAULY M.: Building and animating user-specific volumetric face rigs. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2016).

[IT16] ICHIM A. E., TOMBARI F.: Semantic parametric body shape estimation from noisy depth sequences. *Robotics and Autonomous Systems 75* (2016), 539–549.

[ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2004), pp. 131–140.

[Jac13] JACOBSON A.: *Algorithms and interfaces for real-time deformation of 2d and 3d shapes.* PhD thesis, ETH, 2013.

[JBPS11] JACOBSON A., BARAN I., POPOVIC J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 30*, 4 (2011), 78.

[JEOG11] JIMENEZ J., ECHEVARRIA J. I., OAT C., GUTIERREZ D.: *GPU Pro 2.* AK Peters Ltd., 2011, ch. Practical and Realistic Facial Wrinkles Animation.

[JKSH13] JACOBSON A., KAVAN L., SORKINE-HORNUNG O.: Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 32*, 4 (2013), 33.

[JSP*15] JEONG S., SOLENTHALER B., POLLEFEYS M., GROSS M., ET AL.: Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics 34*, 6 (2015), 199.

[JTPSH15] JAKOB W., TARINI M., PANOZZO D., SORKINE-HORNUNG O.: Instant field-aligned meshes. *Proceedings of ACM SIGGRAPH Asia 34*, 6 (2015), 189:1–189:15.

[KHS03] KÄHLER K., HABER J., SEIDEL H.-P.: Reanimating the dead: reconstruction of expressive faces from skull data. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2003), vol. 22, ACM, pp. 554–561.

[KIL*16] KADLECEK P., ICHIM A.-E., LIU T., KRIVANEK J., KAVAN L.: Reconstructing personalized anatomical models for physics-based body animation. *Proceedings of ACM SIGGRAPH Asia 35*, 6 (2016).

[KKN*13] KIM D., KOH W., NARAIN R., FATAHALIAN K., TREUILLE A., O'BRIEN J. F.: Near-exhaustive precomputation of secondary cloth effects. *ACM Transactions on Graphics 32*, 4 (2013), 87.

[KRP*15] KLEHM O., ROUSSELLE F., PAPAS M., BRADLEY D., HERY C., BICKEL B., JAROSZ W., BEELER T.: Recent advances in facial appearance capture. In *Computer Graphics Forum* (2015), vol. 34, pp. 709–733.

[KSB11] KEMELMACHER-SHLIZERMAN I., BASRI R.: 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011).

[LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. *Proceedings of ACM SIGGRAPH Asia* (2009).

[LAR*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F. H., DENG Z.: Practice and theory of blendshape facial models. In *Eurographics (State of the Art Reports)* (2014), pp. 199–218.

[LBOK13] LIU T., BARGTEIL A. W., O'BRIEN J. F., KAVAN L.: Fast simulation of mass-spring systems. *Proceedings of ACM SIGGRAPH Asia 32*, 6 (2013), 209:1–7.

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 165–172.

[LGK*10] LEE D., GLUECK M., KHAN A., FIUME E., JACKSON K.: A survey of modeling and simulation of skeletal muscle. *ACM Transactions on Graphics 28*, 4 (2010), 1–13.

[LKA*17] LAINE S., KARRAS T., AILA T., HERVA A., SAITO S., YU R., LI H., LEHTINEN J.: Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2017).

[LMB14] LOPER M., MAHMOOD N., BLACK M. J.: Mosh: Motion and shape capture from sparse markers. *Proceedings of ACM SIGGRAPH Asia 33*, 6 (2014), 220.

[LSF12] LLOYD J. E., STAVNESS I., FELS S.: Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, pp. 355–394.

[LSNP13] LI D., SUEDA S., NEOG D. R., PAI D. K.: Thin skin elastodynamics. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 32*, 4 (2013), 49.

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1421–1430.

[LST09] LEE S.-H., SIFAKIS E., TERZOPOULOS D.: Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics 28,* 4 (2009), 99.

[LT06] LEE S.-H., TERZOPOULOS D.: Heads up!: biomechanical modeling and neuromuscular control of the neck. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2006), vol. 25, pp. 1188–1198.

[LT08] LEE S.-H., TERZOPOULOS D.: Spline joints for multibody dynamics. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2008), vol. 27, p. 22.

[LWP10] LI H., WEISE T., PAULY M.: Example-based facial rigging. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2010), vol. 29, ACM, p. 32.

[LXC*15] LI J., XU W., CHENG Z., XU K., KLEIN R.: Lightweight wrinkle synthesis for 3d facial modeling and animation. *Computer-Aided Design 58,* 0 (2015), 117 – 122. Solid and Physical Modeling 2014.

[LYYB13] LI H., YU J., YE Y., BREGLER C.: Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 32,* 4 (2013), 42–1.

[MBF*14] MUNARO M., BASSO A., FOSSATI A., VAN GOOL L., MENEGATTI E.: 3d reconstruction of freely moving persons for re-identification with a depth sensor. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), IEEE, pp. 4512–4519.

[MEAW12] MAAS S. A., ELLIS B. J., ATESHIAN G. A., WEISS J. A.: Febio: finite elements for biomechanics. *Journal of biomechanical engineering 134,* 1 (2012), 011005.

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication and Image Representation 18,* 2 (2007), 109–118.

[MJC*08] MA W.-C., JONES A., CHIANG J.-Y., HAWKINS T., FREDERIKSEN S., PEERS P., VUKOVIC M., OUHYOUNG M., DEBEVEC P.: Facial performance synthesis using deformation-driven polynomial displacement maps. *Proceedings of ACM SIGGRAPH Asia* (2008).

[MKHG13] MALLESON C., KLAUDINY M., HILTON A., GUILLEMAUT J.-Y.: Single-view rgbd-based reconstruction of dynamic human geometry. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on* (2013), IEEE, pp. 307–314.

[ML14] MUJA M., LOWE D. G.: Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36* (2014).

[MLSS94] MURRAY R. M., LI Z., SASTRY S. S., SASTRY S. S.: *A mathematical introduction to robotic manipulation.* CRC press, 1994.

[MSS*17] MEHTA D., SRIDHAR S., SOTNYCHENKO O., RHODIN H., SHAFIEI M., SEIDEL H.-P., XU W., CASAS D., THEOBALT C.: Vnect: Real-time 3d human pose estimation with a single rgb camera. vol. 36.

[MWF*12] MA W.-C., WANG Y.-H., FYFFE G., CHEN B.-Y., DEBEVEC P.: A blendshape model that incorporates physical interaction. *Computer Animation and Virtual Worlds 23*, 3-4 (2012), 235–243.

[MZS*11] MCADAMS A., ZHU Y., SELLE A., EMPEY M., TAMSTORF R., TERAN J., SIFAKIS E.: Efficient elasticity for character skinning with contact and collisions. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 37.

[NFA*15] NAGANO K., FYFFE G., ALEXANDER O., BARBIC J., LI H., GHOSH A., DEBEVEC P. E.: Skin microstructure deformation with displacement map convolution. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015), 109.

[NFS15] NEWCOMBE R. A., FOX D., SEITZ S. M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Computer Vision and Pattern Recognition* (June 2015).

[NIH*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHI P., SHOTTON J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on* (2011), IEEE, pp. 127–136.

[NW06] NOCEDAL J., WRIGHT S.: *Numerical optimization.* Springer Science & Business Media, 2006.

[Oat07] OAT C.: Animated wrinkle maps. In *ACM SIGGRAPH 2007 courses* (2007).

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2003).

[PMRMB15] PONS-MOLL G., ROMERO J., MAHMOOD N., BLACK M. J.: Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015), 120.

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on* (2001), IEEE, pp. 145–152.

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Computer Graphics Forum (Proc. of the EG/SIGGRAPH Symposium on Geometry processing)* (2007), vol. 4.

[SB12] SIFAKIS E., BARBIC J.: Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (2012), p. 20.

[SD92] SHOEMAKE K., DUFF T.: Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface* (1992), vol. 92, Citeseer, pp. 258–264.

[SFCH13] SCHLEIP R., FINDLEY T. W., CHAITOW L., HUIJING P.: *Fascia: the tensional network of the human body: the science and clinical applications in manual and movement therapy.* Elsevier Health Sciences, 2013.

[Sho85] SHOEMAKE K.: Animating rotation with quaternion curves. In *Computer Graphics (Proceedings SIGGRAPH)* (1985), vol. 19, ACM, pp. 245–254.

[Si15] SI H.: Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS) 41*, 2 (2015), 11.

[SLC09] SARAGIH J. M., LUCEY S., COHN J. F.: Face alignment through subspace constrained mean-shifts. In *International Conference on Computer Vision* (2009).

[SLC11] SARAGIH J. M., LUCEY S., COHN J. F.: Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision 91*, 2 (2011), 200–215.

[SLST14] SI W., LEE S.-H., SIFAKIS E., TERZOPOULOS D.: Realistic biomechanical simulation and control of human swimming. *ACM Transactions on Graphics 34*, 1 (2014), 10.

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2005), vol. 24, pp. 417–425.

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2004), vol. 23, pp. 399–405.

[STC*13] SKOURAS M., THOMASZEWSKI B., COROS S., BICKEL B., GROSS M.: Computational design of actuated deformable characters. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 32*, 4 (2013), 82.

[STK*14] SKOURAS M., THOMASZEWSKI B., KAUFMANN P., GARG A., BICKEL B., GRINSPUN E., GROSS M.: Designing inflatable structures. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 33*, 4 (2014), 63.

[SWH*16] SAITO S., WEI L., HU L., NAGANO K., LI H.: Photorealistic facial texture inference using deep neural networks. *arXiv preprint arXiv:1612.00523* (2016).

[SWTC14] SHI F., WU H.-T., TONG X., CHAI J.: Automatic acquisition of high-fidelity facial performances using monocular videos. *Proceedings of ACM SIGGRAPH Asia 33*, 6 (2014), 222.

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2005), vol. 24, pp. 488–495.

[SZK15] SAITO S., ZHOU Z.-Y., KAVAN L.: Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015).

[TBHF03] TERAN J., BLEMKER S., HING V., FEDKIW R.: Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2003), Eurographics Association, pp. 68–74.

[The16] THE CGAL PROJECT: *CGAL User and Reference Manual*, 4.8 ed. CGAL Editorial Board, 2016.

[TMB14] TSOLI A., MAHMOOD N., BLACK M. J.: Breathing life into shape: capturing, modeling and animating 3d human breathing. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 33*, 4 (2014), 52.

[TSB*05] TERAN J., SIFAKIS E., BLEMKER S. S., NG-THOW-HING V., LAU C., FEDKIW R.: Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 317–328.

[TSIF05] TERAN J., SIFAKIS E., IRVING G., FEDKIW R.: Robust quasistatic finite elements and flesh simulation. In *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2005), ACM.

[TSSP16] TOMPSON J., SCHLACHTER K., SPRECHMANN P., PERLIN K.: Accelerating eulerian fluid simulation with convolutional networks. *arXiv preprint arXiv:1607.03597* (2016).

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models.

[VCL*06] VANDERMEULEN D., CLAES P., LOECKX D., DE GREEF S., WILLEMS G., SUETENS P.: Computerized craniofacial reconstruction using ct-derived implicit surface representations. *Forensic science international 159* (2006), S164–S174.

[vdPJD*14] VON DER PAHLEN J., JIMENEZ J., DANVOYE E., DEBEVEC P., FYFFE G., ALEXANDER O.: Digital ira and beyond: Creating real-time photoreal digital actors. In *ACM SIGGRAPH 2014 Courses* (2014), SIGGRAPH '14.

[VLR05] VENKATARAMAN K., LODHA S., RAGHAVAN R.: A kinematic-variational model for animating skin with wrinkles. *Computers & Graphics* (2005).

[VWB*12] VALGAERTS L., WU C., BRUHN A., SEIDEL H.-P., THEOBALT C.: Lightweight binocular facial performance capture under uncontrolled lighting. *Proceedings of ACM SIGGRAPH Asia* (2012).

[WB06] WÄCHTER A., BIEGLER L. T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming 106*, 1 (2006).

[WBG*16a] WU C., BRADLEY D., GARRIDO P., ZOLLHÖFER M., THEOBALT C., GROSS M., BEELER T.: Model-based teeth reconstruction. *ACM Transactions on Graphics 35*, 6 (2016), 220.

[WBG*16b] WU C., BRADLEY D., GARRIDO P., ZOLLHÖFER M., THEOBALT C., GROSS M., BEELER T.: Model-based teeth reconstruction. *Proceedings of ACM SIGGRAPH Asia 35*, 6 (2016).

[WBGB16] WU C., BRADLEY D., GROSS M., BEELER T.: An anatomically-constrained local deformation model for monocular face capture. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 35*, 4 (2016), 115.

[WBLP11] WEISE T., BOUAZIZ S., LI H., PAULY M.: Realtime performance-based facial animation. In *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2011), vol. 30, ACM, p. 77.

[WET13] WETA DIGITAL: Tissue system. http://www.fxguide.com/fxguidetv/fxguidetv-166-weta-digitals-tissue-system/, 2013.

[WHB11] WEISS A., HIRSHBERG D., BLACK M. J.: Home 3d body scans from noisy image and range data. In *International Conference on Computer Vision* (2011), IEEE, pp. 1951–1958.

[WHC*16] WEI L., HUANG Q., CEYLAN D., VOUGA E., LI H.: Dense human body correspondences using convolutional networks. In *Computer Vision and Pattern Recognition* (2016).

[WKT96] WU Y., KALRA P., THALMANN N. M.: Simulation of static and dynamic wrinkles of skin. In *Proc. of IEEE Computer Animation* (1996).

[WLVGP09] WEISE T., LI H., VAN GOOL L., PAULY M.: Face/off: Live facial puppetry. *Proceedings of the EG/SIGGRAPH Symposium on Computer Animation* (2009).

[WMG96] WEISS J. A., MAKER B. N., GOVINDJEE S.: Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer methods in applied mechanics and engineering 135*, 1 (1996), 107–128.

[WSA*02] WU G., SIEGLER S., ALLARD P., KIRTLEY C., LEARDINI A., ROSENBAUM D., WHITTLE M., D D'LIMA D., CRISTOFOLINI L., WITTE H., ET AL.: Isb recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part i: ankle, hip, and spine. *Journal of biomechanics 35*, 4 (2002), 543–548.

[Wu13] WU C.: Towards linear-time incremental structure from motion. In *3D Vision, 2013 International Conference on* (June 2013).

[WVdHV*05] WU G., VAN DER HELM F. C., VEEGER H. D., MAKHSOUS M., VAN ROY P., ANGLIN C., NAGELS J., KARDUNA A. R., MCQUADE K., WANG X., ET AL.: Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—part ii: shoulder, elbow, wrist and hand. *Journal of biomechanics 38*, 5 (2005), 981–992.

[WWY*15] WANG B., WU L., YIN K., ASCHER U., LIU L., HUANG H.: Deformation capture and modeling of soft objects. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 34*, 4 (2015), 94.

[WZN*14] WU C., ZOLLHÖFER M., NIESSNER M., STAMMINGER M., IZADI S., THEOBALT C.: Proceedings of acm siggraph asia. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 200:1–200:10.

[YCP16] YEUNG Y.-H., CROUCH J., POTHEN A.: Interactively cutting and constraining vertices in meshes using augmented matrices. *ACM Transactions on Graphics 35*, 2 (Feb. 2016), 18:1–18:17.

[YY14] YE M., YANG R.: Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Computer Vision and Pattern Recognition* (2014).

[ZB15] ZUFFI S., BLACK M. J.: The stitched puppet: A graphical model of 3d human shape and pose. In *Computer Vision and Pattern Recognition* (2015), pp. 3537–3546.

[ZE00] ZOMORODIAN A., EDELSBRUNNER H.: Fast software for box intersections. In *Proceedings of the sixteenth annual symposium on Computational geometry* (2000), ACM, pp. 129–138.

[ZHK15] ZHU L., HU X., KAVAN L.: Adaptable anatomical models for realistic bone motion reconstruction. *Computer Graphics Forum 34*, 2 (2015).

[ZNI*14] ZOLLHÖFER M., NIESSNER M., IZADI S., REHMANN C., ZACH C., FISHER M., WU C., FITZGIBBON A., LOOP C., THEOBALT C., STAMMINGER M.: Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics 33*, 4 (2014).

[ZPB07] ZACH C., POCK T., BISCHOF H.: A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*. Springer, 2007, pp. 214–223.

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics* (2004).

[Zyg16] ZYGOTE: Zygote body, 2016. [Online; accessed 28-Dec-2016].

Alexandru Eugen Ichim

École Polytechnique Fédérale de Lausanne
+41 78 656 29 95   alex.e.ichim@gmail.com
http://lgg.epfl.ch/~ichim/

Education

05/2013 - to date
École Polytechnique Fédérale de Lausanne, Switzerland
PhD student in the Computer Graphics and Geometry Laboratory under the supervision of Prof. Dr. Mark Pauly. Focusing on digital human reconstruction and performance capture. Expected to graduate in 2017.

09/2011 - 04/2013
École Polytechnique Fédérale de Lausanne, Switzerland
Master of Science in Computer Science. GPA: 5.7/6 (on a scale from 1 to 6) in 92 ECTS credits.
Thesis: *RGB-D Handheld Mapping and Modeling*, supervised by Dr. Radu Rusu and Prof. Dr. Mark Pauly, received maximum grade of 6.0

09/2008 - 06/2011
Jacobs University, Bremen, Germany
Bachelor of Science in Electrical Engineering and Computer Science, specializing in Computer Science. GPA: 1.30 (on a scale from 5 to 1) in 202.5 ECTS credits.
Thesis: *Path Planning in 3D Unstructured Environments*, supervised by Prof. Dr. Andreas Birk, received maximum grade of 1.0

09/2004 - 07/2008
"Gheorghe Vranceanu" National College, Bacau, Romania
Graduated in top 1% in the Mathematics, intensive Computer Science Class.

Publications

Phace: Physics-based Face Modeling and Animation
Alexandru-Eugen Ichim, Petr Kadlecek, Ladislav Kavan, Mark Pauly
ACM Transactions on Graphics, Proceedings of SIGGRAPH 2017

Reconstructing Personalized Anatomical Models for Physics-based Body Animation
Petr Kadlecek*, Alexandru-Eugen Ichim*, Tiantian Liu, Ladislav Kavan, Jaroslav Krivanek
ACM Transactions on Graphics, Proceedings of SIGGRAPH ASIA 2016, (*joint first authors)

Building and Animating User-Specific Volumetric Face Rigs
Alexandru-Eugen Ichim, Ladislav Kavan, Merlin Nimier-David, Mark Pauly
ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA 2016

Patient MoCap: Human Pose Estimation under Blanket Occlusion for Hospital Monitoring Applications
Felix Achilles, Alexandru-Eugen Ichim, Huseyin Coskun, Federico Tombari, Soheyl Noachtar, Nassir Navab
International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI 2016

Temporally Consistent Motion Segmentation from RGB-D Video
Peter Bertholet, Alexandru-Eugen Ichim, Matthias Zwicker. arXiv 2016

Dynamic 3D avatar creation from hand-held video input
Alexandru-Eugen Ichim, Sofien Bouaziz, Mark Pauly
ACM Transactions on Graphics, Proceedings of SIGGRAPH 2015

Semantic parametric body shape estimation from noisy RGB-D sequences
Alexandru-Eugen Ichim, Federico Tombari
Robotics and Autonomous Systems, IEEE RAS 2015

A modular framework for aligning 3D point clouds - registration with the Point Cloud Library
Dirk Holz, Alexandru-Eugen Ichim, Federico Tombari, Radu B. Rusu, Sven Behnke
Robotics & Automation Magazine, IEEE RAM 2015

The Jacobs Robotics approach to object recognition and localization in the context of the ICRA'11 solutions in perception challenge
Narunas Vaskevicius, Kaustubh Pathak, Alexandru-Eugen Ichim, Andreas Birk
IEEE International Conference on Robotics and Automation, ICRA 2012

DSim. A Tool for Assisted Spatial Design
Mehul Bhatt, Gregory Flanagan, Alexandru-Eugen Ichim
4th International Conference on Design Computing and Cognition, DCC 2010

Reviewer for multiple Computer Graphics, Computer Vision and Robotics publications (2013-2017):
SIGGRAPH, SIGGRAPH ASIA, Eurographics, Pacific Graphics, SCA, IEEE VR, CASA, RAS, RAM, ICRA

Practical Experience

07/2016 - 10/2016	**Adobe Systems, Inc., Creative Technologies Lab, Seattle, USA** Research intern working on a novel physics framework for enforcing inter-layer relationships (e.g., contact, example-based) for artist-directed multi-layer 2D animation. Supervised by Dr. Jovan Popović and Dr. Danny Kaufman.
03/2012 - to date	**Open Perception, Inc., Point Cloud Library** Research scientist and software project maintainer, involves offering user support on the mailing lists/forums of the project, solving administrative issues such as keeping the build farm running, fixing code bugs, announcing news, advertising the project etc. Administered various code sprints and the Google Summer of Code 2012 and 2014 programs. Tasks involve interfacing the organization with the sponsors and coordinating the student selection and project work. Several code sprints with external companies, such as Toyota. Worked on surface reconstruction and scene understanding via superquadric fitting.
08/2012 - 04/2013	**Willow Garage, Inc., Menlo Park, USA** Intern under the supervision of Dr. Radu Rusu. Working on 3D object modeling and indoor reconstruction with RGB-D cameras using only depth information and planar features, as part of my master thesis project.
09/2011 - 07/2012	**Computer Graphics and Geometry Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland** Student research assistant in the group of Prof. Dr. Mark Pauly, in collaboration with Faceshift. Working on improving 3D face and expression tracking using affordable 3D cameras such as the Microsoft Kinect or the Asus Xtion Pro.
06/2011 - 09/2011	**Google Summer of Code Program 2011, Point Cloud Library** Participated in writing and improving algorithms in the Point Cloud Library, under the supervision of Dr. Radu Rusu. Concentrated more on 3D feature extraction, but topics also included: point cloud smoothing, interest region / keypoint detection, registration, (hand gesture) classification.
06/2010 - 08/2010	**Computer Graphics and Geometry Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland** Research intern in the group of Prof. Dr. Mark Pauly. Worked on an application for facial tracking (3D rigid body transform and expressions) from low-quality monocular video input. Involved experimentation with various computer vision and computer graphics techniques and concepts.
06/2009 - 06/2011	**Jacobs Robotics Research Group, Jacobs University, Bremen, Germany** Student research assistant for the Co3 AUVs EU project (Cooperative Cognitive Control for Autonomous Underwater Vehicles) and the RobLog EU project, supervised by Prof. Dr. Andreas Birk. Created a framework for processing point/plane clouds integrating algorithms for cloud matching, plane extraction etc. Developed the machine vision module for the SICK Robot Day 2009 contest and integrated it in the Jacobs rescue robot. Programmed the GUI for interacting with the real and virtual robots and other components for the RoboCup 2009 competition. Designed and implemented a 3D underwater robotics simulator for distributed computing. The application is used for teaching and research by all the partners of the Co3 AUVs project.
06/2009 - 08/2009	**Deutsches Forschungszentrum für Künstliche Intelligenz (DKFI) Bremen – Robotics Innovation Center** Summer internship. Developed Linux drivers and interfaces for various hardware (GPS board, SwissRanger SR3000, firewire cameras). Created an application that would interact with the SR3000 for gathering and processing point clouds and researched and evaluated various SLAM algorithms.
01/2009 - 11/2009	**Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) Bremen - Cognitive Systems** Worked on the DSim project in collaboration with Dr. Mehul Bhatt. Built a 3D simulator for an indoor world where avatars interact with the environment through various types of sensors.
09/2009 - 12/2009	**Independent Study Course - Point Cloud data visualization and processing – Visualization and Computer Graphics Laboratory, Jacobs University, Bremen, Germany** Analyzed novel ways of improving the visualization and matching of point clouds using strictly computer graphics specific approaches.

Teaching Experience

05/2013 - to date **École Polytechnique Fédérale de Lausanne, Switzerland**

Teaching Assistant for the following courses:

- Digital Geometry Processing – Spring 2014, Spring 2015, Fall 2016
- Advanced Computer Graphics – Fall 2014, Fall 2015
- Introduction to Computer Graphics – Spring 2016, Spring 2017

Supervised multiple master-level semester projects on topics such as: biomechanical simulations, physical face animation with a color projector, laser line scanner, multiview-stereo facial reconstruction, 3D reconstruction using handheld devices, hair physics simulation.

Presentations and Talks

05/2013 - to date Various presentations and demos organized at EPFL for university-wide events.
Press coverage for our SIGGRAPH 2015 project on avatar creation.

06/2016 PCL Hackfest Summer 2016

07/2013 Half Day Course on the Point Cloud Library
Smart Libraries for Computer Graphics Summer School (CGLibs) 2013, Pisa, Italy

05/2013 Invited talk, GPU Technology Conference, 2013, San Jose, California, USA

11/2011 PCL Tutorial at International Conference on Computer Vision (ICCV) 2011, Barcelona, Spain

Skills and Achievements

Languages: Romanian: mother tongue English: fluent
French: intermediate German: basic knowledge

PC Knowledge: Extensive knowledge of Linux operating systems, Microsoft Windows
Frequent user of: MathWorks MATLAB, Wolfram Research Mathematica, Microsoft Visual Studio, Microsoft SQL Server, Blender, Autodesk Maya, Adobe Photoshop, Premiere, Microsoft Office, LaTeX, ArchiCAD

Technical Skills: Proficient in C/ C++/ C#, Java.
Frameworks: Qt, OpenCV, PCL, Boost, OpenGL, Bullet and Ogre3D
Professional tools: Autodesk Maya, The Foundry Modo, Adobe Premiere, Adobe Photoshop, MS Office
Experience in working with: Python, SQL, HTML, XML, ASP.NET, AJAX, Standard ML, Basic, Ruby, SML

Awards:

- Special Mention for the Student of the Year Excellence Award from LSRS (The League of Romanian Students Abroad), 2012
- 2nd prize at the ICRA 2011 Solutions in Perception Challenge with the Jacobs University Robotics Group
- Jacobs University President's List Award for outstanding academic achievements in all the academic years: 2008-2009, 2009-2010 and 2010-2011
- Various prizes in the National Olympiad of Computer Science, as well as other national and regional physics and mathematics contests – 2004-2008

Interests: Photography, biking, reading, audio geek, entrepreneurship

References

Prof. Mark Pauly
mark.pauly@epfl.ch

Dr. Sofien Bouaziz
sofien.bouaziz@gmail.com

Dr. Jovan Popović
jovan@adobe.com

Dr. Radu B. Rusu
rbrusu@fyusion.com

Prof. Dr. Ladislav Kavan
ladislav.kavan@gmail.com