High-quality Face Capture, Animation and Editing from Monocular Video

A dissertation submitted towards the degree
Doctor of Engineering
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Pablo Garrido

Saarbrücken
April, 2017

UNIVERSITÄT
DES
SAARLANDES

Dean of the Faculty:

Univ.-Prof. Dr. Frank-Olaf Schreyer
Saarland University
Saarbrücken, Germany

Defense:

June 26, 2017, in Saarbrücken

Chair of the Committee:

Prof. Dr. Jürgen Steimle

Examiners:

Prof. Dr. Christian Theobalt
Dr. Patrick Pérez
Prof. Dr. Mark Pauly

Academic Assistant:

Dr. Rhaleb Zayer

Acknowledgments

Foremost, I would like to thank my supervisor Christian Theobalt that provided me with the best possible advice and guidance during my Ph.D. despite his busy agenda. I also want to give special thanks to my closest collaborators at MPI for Informatics, Michael Zollhöfer and Levi Valgaerts, with whom I spent long hours discussing ideas, algorithms, and non-work related matters. I also take the opportunity to thank one of my closest external collaborators, Patrick Pérez, who always provided exceptional feedback in my online meetings and spent lots of hours reviewing my papers and thesis. I am also more than grateful to my parents and especially my wife that unconditionally stood by my side, always supported all my decisions, and gave me the strength to keep pushing towards my goals.

It was truly an honor to have worked in the Computer Graphics Department at MPI for Informatics, headed by Prof. Dr. Hans-Peter Seidel. The quality of the research is just remarkable, and so are the researchers in it. Past and current members of the GVV Group made my stay at MPI for Informatics an enjoyable experience that I will never forget. Especially, I would like to mention my dear colleagues Helge Rhodin, Hyeongwoo Kim, Nadia Robertini, Srinath Sridhar, Miguel Granados, Chenglei Wu, Ayush Tewari, Dushyant Mehta, Franziska Müller, and Abhimitra Meka. Here, I could also not forget the administrative staff members of the Computer Graphics Department, Sabine Budde and Ellen Fries, who were always very responsive and attended to all my requests.

During my Ph.D., I had the great opportunity to work with talented researchers at Technicolor, Disney Research, Adobe Research, and Saarland University. I am particularly grateful to Patrick Pérez, Thabo Beeler, Derek Bradley, Kalyan Sunkavalli, and Ingmar Steiner for having shared their expert knowledge while developing the different projects.

I am indebted to funding agencies that financed my postgraduate studies and Ph.D. thesis: the Deutscher Akademischer Austauschdienst and the Max Planck Society.

Finally, I would like to thank all my relatives and friends that despite the distance they always cheered me up and gave lots of support.

Abstract

Digitization of virtual faces in movies requires complex capture setups and extensive manual work to produce superb animations and video-realistic editing. This thesis pushes the boundaries of the digitization pipeline by proposing automatic algorithms for high-quality 3D face capture and animation, as well as photo-realistic face editing. These algorithms reconstruct and modify faces in 2D videos recorded in uncontrolled scenarios and illumination. In particular, advances in three main areas offer solutions for the lack of depth and overall uncertainty in video recordings. First, contributions in capture include model-based reconstruction of detailed, dynamic 3D geometry that exploits optical and shading cues, multilayer parametric reconstruction of accurate 3D models in unconstrained setups based on inverse rendering, and regression-based 3D lip shape enhancement from high-quality data. Second, advances in animation are video-based face reenactment based on robust appearance metrics and temporal clustering, performance-driven retargeting of detailed facial models in sync with audio, and the automatic creation of personalized controllable 3D rigs. Finally, advances in plausible photo-realistic editing are dense face albedo capture and mouth interior synthesis using image warping and 3D teeth proxies. High-quality results attained on challenging application scenarios confirm the contributions and show great potential for the automatic creation of photo-realistic 3D faces.

Kurzzusammenfassung

Die Digitalisierung von Gesichtern zum Einsatz in der Filmindustrie erfordert komplizierte Aufnahmevorrichtungen und die manuelle Nachbearbeitung von Rekonstruktionen, um perfekte Animationen und realistische Videobearbeitung zu erzielen. Diese Dissertation erweitert vorhandene Digitalisierungsverfahren durch die Erforschung von automatischen Verfahren zur qualitativ hochwertigen 3D Rekonstruktion, Animation und Modifikation von Gesichtern. Diese Algorithmen erlauben es, Gesichter in 2D Videos, die unter allgemeinen Bedingungen und unbekannten Beleuchtungsverhältnissen aufgenommen wurden, zu rekonstruieren und zu modifizieren. Vorallem Fortschritte in den folgenden drei Hauptbereichen tragen zur Kompensation von fehlender Tiefeninformation und der allgemeinen Mehrdeutigkeit von 2D Videoaufnahmen bei. Erstens, Beiträge zur modellbasierten Rekonstruktion von detaillierter und dynamischer 3D Geometrie durch optische Merkmale und die Shading-Eigenschaften des Gesichts, mehrschichtige parametrische Rekonstruktion von exakten 3D Modellen mittels inversen Renderings in allgemeinen Szenen und regressionsbasierter 3D Lippenformverfeinerung mittels qualitativ hochwertigen Daten. Zweitens, Fortschritte im Bereich der Computeranimation durch videobasierte Gesichtsausdrucksübertragung und temporaler Clusterbildung, Übertragung von detaillierten Gesichtsmodellen, deren Mundbewegung mit Ton synchronisiert ist, und die automatische Erstellung von personalisierten “3D Face Rigs”. Schließlich werden Fortschritte im Bereich der realistischen Videobearbeitung vorgestellt, welche auf der dichten Rekonstruktion von Hautreflektionseigenschaften und der Mundinnenraumsynthese mittels bildbasierten und geometriebasierten Verfahren aufbauen. Qualitativ hochwertige Ergebnisse in anspruchsvollen Anwendungen untermauern die Wichtigkeit der geleisteten Beiträgen und zeigen das große Potential der automatischen Erstellung von realistischen digitalen 3D Gesichtern auf.

Summary

Thanks to cutting-edge advances in technology in the fields of Computer Graphics and Vision in the last two decades, the entertainment industry is now capable of bringing digital models of our favorite actors to life in a very realistic way. The movie industry employs such technology for a range of purposes, from the complexity of live action shots to the need for photo-realistic virtual characters that resemble an actor's appearance, e. g., a younger or older digital double. Digitizing photo-realistic humans, especially faces, is a highly complex process. It is difficult because the human eye is accustomed to identifying faces in our daily lives, and thus, our expert eye can easily spot even the smallest inaccuracies in digital models. To achieve the desired level of photorealism, the digitization pipeline in post-production performs several standardized technical steps. First, it creates a high-quality fully-controllable 3D model in shape and appearance of the actor's face – often referred to as a "face rig" in the literature – to be animated by skilled digital artists. Then, the model is rendered under desired lighting conditions. Finally, it is inserted back into the scene in an editing step to create the final composite. We know that these steps to reconstruct detailed personalized 3D face models and accurate facial motion require sophisticated capture setups and studio controlled illumination to achieve the animation of photo-realistic digital faces; thus, it remains a challenge to successfully utilize the digitization pipeline. Moreover, this pipeline relies primarily on the expertise of an artist. He must manually improve both the face models and the facial animations to make them look realistic when rendered back into the video – an effortful, lengthy and tedious task.

This thesis is motivated by the limitations in the capture process and the great deal of manual work in the digitization pipeline. We develop robust and fully automatic algorithms that push the boundaries of digitization further and that aim to capture highly-detailed animated 3D face models and photo-realistically modify faces with these models in unconstrained 2D footage recorded under uncontrolled lighting. Note that the algorithms' tasks are ambitious due to the lack of 3D information and overall uncertainty in a scene, e. g., (self) occlusions, sudden and expressive facial motion, lighting changes, and out-of-plane head rotation. What this thesis does is present automatic and accurate model-based methods for capturing highly-detailed facial performances, animating controllable 3D facial models at high fidelity, and editing photo-realistic faces with plausible mouth interior. These methods all unify in a framework that improves on the underlying representation of the face to handle more challenging video input and perform more advanced editing tasks. As a proof of concept, we test our proposed methods on different real-life application scenarios, including face reenactment, dubbing, face modification, and video rewriting.

The technical contributions of this thesis can be divided into three main areas: capture, animation, and editing.

Capture The main improvements over state-of-the-art approaches can be summarized as follows: Chapter 4 presents an accurate approach that refines 2D facial landmark locations using optical flow between automatically selected keyframes. Such 2D landmarks are used later to

assist the tracking of 3D face models. Then, Chapter 5 introduces a drift-free model-based tracking approach based on accurate 2D landmarks as well as dense optical and shading cues in the temporal domain to obtain detailed, dynamic 3D geometry and estimate the incident lighting in semi-constrained video sequences. To improve tracking further, Chapter 7 proposes a robust fully-parametric face capture method that inverts the image formation model to reconstruct multiple layers of personalization and details from unconstrained 2D footage, e.g., a YouTube video. Finally, Chapter 9 demonstrates an effective data-driven lip regression approach that leverages a new database of high-quality multiview reconstructions to enable high-quality 3D lip shape reconstruction even from monocular video input.

Animation The main contributions in this area are mainly concerned with novel retargeting and modeling techniques relevant to facial animation. Chapter 4 presents a simple, yet effective, video-based approach that transfers temporally-coherent facial expressions between two arbitrary performances by leveraging robust appearance and motion descriptors, as well as hierarchical clustering, to preserve temporal consistency. Chapter 6 introduces a system for performance-driven model-based retargeting and resynthesis of detailed facial models that can also align the optical channel with an audio signal for visual dubbing. Finally, Chapter 8 demonstrates that personalized high-quality 3D face rigs, which generate new person-specific expressions and details by simply modifying intuitive motion controllers, can be created from unconstrained monocular performances.

Editing The main contributions in this area are summarized as follows: Chapter 4 presents a method for synthesizing a plausible mouth interior using simple image warping techniques. Chapter 6 improves upon this simple approach by adding a 3D teeth proxy. It also shows a method for capturing realistic dense face albedo that, when combined with the estimated scene lighting (Chapter 5), can render photo-realistic 3D face models back into the original video, as demonstrated in Chapters 6 and 8.

To summarize, this thesis presents several robust and automatic algorithms that aim at capturing, animating, and editing photo-realistic synthetic face models at high fidelity from arbitrary 2D video and that are affordable for anyone. The proposed scientific contributions greatly advance the state of the art in monocular facial performance capture and face capture-based video editing, thus enormously improving the toolbox available for creating photo-realistic human face avatars from 2D video footage. Results attained on different application scenarios show great potential to automatize the digitization of photo-realistic virtual characters in movies and games, and possibly virtual communication, in the near future.

Contents

1 Introduction **1**
- 1.1 Topic and Motivation . . . 1
- 1.2 Scope and Overview . . . 3
- 1.3 Structure . . . 4
  - 1.3.1 Summary of Technical Chapters . . . 5
- 1.4 Technical Contributions . . . 7
- 1.5 List of Publications . . . 8

2 Basics **9**
- 2.1 Facial Animation and Modeling . . . 9
  - 2.1.1 Blendshapes . . . 9
  - 2.1.2 Facial Rig . . . 11
- 2.2 Camera and Image Formation Model . . . 11
  - 2.2.1 Camera Model . . . 11
  - 2.2.2 Image Formation Model . . . 14

3 Related Work **17**
- 3.1 Facial Performance Capture . . . 17
  - 3.1.1 Dense Facial Performance Capture . . . 17
  - 3.1.2 Lightweight Facial Performance Capture . . . 21
  - 3.1.3 Monocular Facial Performance Capture . . . 23
- 3.2 Lip Tracking and Reconstruction . . . 29
  - 3.2.1 Image-based 2D Contour Tracking . . . 29
  - 3.2.2 Dense 3D Lip Reconstruction . . . 31
- 3.3 Face Rig and Detail Generation . . . 32
- 3.4 Speech-driven and Video-driven Facial Animation . . . 34
  - 3.4.1 Speech-driven Animation . . . 35
  - 3.4.2 Video-driven Facial Animation . . . 36
- 3.5 Face Replacement and Rewriting in Video . . . 39
  - 3.5.1 Face Replacement . . . 40
  - 3.5.2 Face Rewriting . . . 41

4 Image-based Face Capture and Reenactment **45**
- 4.1 Introduction . . . 46
- 4.2 Overview . . . 47
- 4.3 Non-rigid Face Tracking . . . 48
  - 4.3.1 Automatic Key Frame Selection . . . 48
  - 4.3.2 Optical Flow-based Feature Correction . . . 50
- 4.4 Face Matching . . . 51

4.4.1 Image Alignment and Feature Extraction . . . 52
4.4.2 Temporal Clustering and Frame Selection . . . 52
4.5 Face Transfer . . . 55
4.5.1 Shape and Appearance Transfer . . . 56
4.6 Experiments . . . 58
4.6.1 Results . . . 60
4.6.2 Validations . . . 61
4.7 Discussion and Limitations . . . 64
4.8 Summary . . . 65

5 Model-based Face Capture in Semi-Constrained Setups 67
5.1 Introduction . . . 68
5.2 Overview . . . 69
5.3 Personalized Blendshape Model Creation . . . 70
5.4 Blendshape Tracking . . . 70
5.4.1 Accurate 2D Facial Feature Tracking . . . 70
5.4.2 Coarse Expression and Rigid Pose Estimation . . . 72
5.5 Dense Tracking Correction . . . 74
5.5.1 Temporally Coherent Corrective Flow . . . 74
5.5.2 Optical Flow-based Mesh Deformation . . . 77
5.6 Dynamic Shape Refinement . . . 77
5.7 Experiments . . . 78
5.7.1 Results . . . 78
5.7.2 Validation . . . 82
5.8 Discussion and Limitations . . . 83
5.9 Summary . . . 84

6 Model-based Face Retargeting: A Visual Dubbing Approach 85
6.1 Introduction . . . 86
6.2 Background: Visual Cues in Speech Perception . . . 87
6.3 Overview . . . 87
6.4 Motion Transfer . . . 89
6.4.1 Monocular Facial Performance Capture . . . 89
6.4.2 Blendshape Weight-based Mouth Transfer . . . 89
6.4.3 Mouth Motion Correction . . . 91
6.5 Detail Synthesis . . . 91
6.5.1 Target Frame Retrieval: Energy Formulation . . . 91
6.5.2 Target Frame Retrieval: Energy Optimization . . . 93
6.5.3 Analysis of Energy Terms and Parameter Tuning . . . 94
6.5.4 Detail Transfer . . . 94
6.6 Speech Alignment . . . 95
6.7 Rendering and Compositing . . . 96
6.7.1 Rendering the Synthesized Geometry . . . 96
6.7.2 Teeth, Inner Mouth and Final Composite . . . 96
6.8 Experiments . . . 97
6.8.1 Results . . . 97
6.8.2 Validations . . . 99
6.9 Discussion and Limitations . . . 102
6.10 Summary . . . 104

7 Multilayer Model-based Face Capture in Unconstrained Setups 105
7.1 Introduction . . . 106
7.2 Overview . . . 107
7.3 Multilayer Personalized 3D Face Prior . . . 108
7.3.1 Camera Parametrization . . . 108
7.3.2 Lighting and Appearance Model . . . 109
7.3.3 Coarse-scale Identity and Expression Model . . . 110
7.3.4 Medium-scale Corrective Shapes . . . 110
7.3.5 Fine-scale Detail Layer . . . 111
7.4 Coarse- and Medium-scale Layer Reconstruction . . . 112
7.4.1 Energy Minimization . . . 112
7.5 Fine-scale Layer Reconstruction . . . 114
7.6 Multi-step Optimization Strategy . . . 115
7.7 Experiments . . . 117
7.7.1 Qualitative and Quantitative Results . . . 118
7.7.2 Validations: Comparison to Performance Capture Approaches . . . 118
7.8 Discussion and Limitations . . . 124
7.9 Summary . . . 125

8 Beyond Face Capture: Face Rig Creation, Animation and Editing 127
8.1 Introduction . . . 128
8.2 Overview . . . 129
8.3 Face Rig Learning . . . 130
8.3.1 Affine Parameter Regression of Correctives and Details . . . 130
8.3.2 Sparse Affine Regression of Fine-scale Details . . . 131
8.4 Face Rig Synthesis . . . 132
8.4.1 Medium-scale Correctives Synthesis . . . 132
8.4.2 Fine-scale Detail Variation Synthesis . . . 132
8.5 Experiments . . . 133
8.5.1 Application Scenarios . . . 133
8.5.2 Validations . . . 135
8.6 Discussion and Limitations . . . 138
8.7 Summary . . . 140

9 Beyond Face Capture: Accurate Lip Tracking 141
9.1 Introduction . . . 142
9.2 Overview . . . 143
9.3 Data Collection . . . 144
9.3.1 High-quality Lip Database . . . 144
9.3.2 Training Data for Regression . . . 145
9.4 Lip Correction Layer Parametrization . . . 147
9.4.1 Dense Correspondence Association . . . 147
9.4.2 Gradient-based Lip Shape Representation . . . 148
9.5 Lip Shape Regression . . . 149
9.5.1 Robust Features for Lip Shape Regression . . . 149
9.5.2 Local Radial Basis Function Networks . . . 150
9.6 Experiments . . . 150
9.6.1 Results . . . 151
9.6.2 Validations . . . 154

9.7 Discussion and Limitations . . . 158
9.8 Summary . . . 159

10 Conclusion **161**
10.1 Summary and Discussion . . . 162
10.2 Extensions . . . 164
10.2.1 Realtime Performance Capture . . . 164
10.2.2 Beyond Face Capture: Model-based Teeth Reconstruction . . . 165
10.3 Future Work and Outlook . . . 166
10.3.1 Challenges in Face Capture . . . 166
10.3.2 Beyond Face Capture: Tongue, Eyes, and Hair Reconstruction . . . 167
10.4 Closing Remarks . . . 169

Appendices **171**

A Multilayer Model-based Face Capture in Unconstrained Setups **173**
A.1 Test Sequences: Description and Specifications . . . 173
A.2 Energy Function: Derivatives . . . 175
A.2.1 Data Objective . . . 175
A.2.2 Prior Objective and Boundary Constraint . . . 179

B Beyond Face Capture: Accurate Lip Tracking **181**
B.1 High-quality Lip Database: Training Examples . . . 181

Chapter 1

Introduction

1.1 Topic and Motivation

Advances in technology in the digitization pipeline now allow the entertainment industry to create and animate digital 3D faces of actors in a very realistic way. The movie industry employs such technology for a range of purposes, from the complexity of live action shots to the need for photo-realistic virtual characters that resemble the appearance of an actor. Some examples that illustrate the use of digital human faces in movies are shown in Figure 1.1.

To achieve the desired level of photorealism in digital scenes, it is mandatory to create custom, photo-realistic face models with personalized expressions and idiosyncrasies that look indistinguishable from the real actor when played on the screen. To this end, post-production has engineered a pipeline that usually comprises four standardized stages: Face rig creation, animation, rendering, and compositing. In the first step, a high-quality, actor-specific 3D face model is captured in professional indoor setups [Klehm et al. 2015]. This personalized model usually contains hundreds of detailed facial expressions performed by the actor, which are then manually improved by digital artists. Then, the artists create deformation mechanisms and interactive high-level motion controllers that activate different facial expressions. This process is called rigging and is done by artists through motion rigs or blendshapes [Komorowski et al. 2010]. In the second step, the face rig is animated either by manually moving the motion controllers or through motion capture data [Beeler et al. 2011; Bickel et al. 2007; Bhat et al. 2013; Bradley et al. 2010; Weise et al. 2009]. In the third step, the 3D facial animations are rendered under desired lighting conditions. Finally, the renderings are blended in with the background scene to create the final composite.

The key to attaining high-quality results in this pipeline is the capture step that requires sophisticated scanning systems [Huang et al. 2004; Weise et al. 2009; Wang et al. 2004] or multiview camera setups [Beeler et al. 2011; Beeler and Bradley 2014] with studio controlled indoor illumination, e. g., light stages [Alexander et al. 2010; Alexander et al. 2013]. Such setups allow for capturing high-quality face albedo and detailed 3D face geometry, including wrinkles and skin pores. However, they are expensive and very hard to build and utilize by non-professional users. Furthermore, each step in the pipeline relies on the expertise of digital artists. They must manually improve the face models and the facial animations, as well as verify the quality of the renderings and the compositing to ensure error-free video animations that do not fall into the uncanny valley – an effortful, lengthy and tedious task.

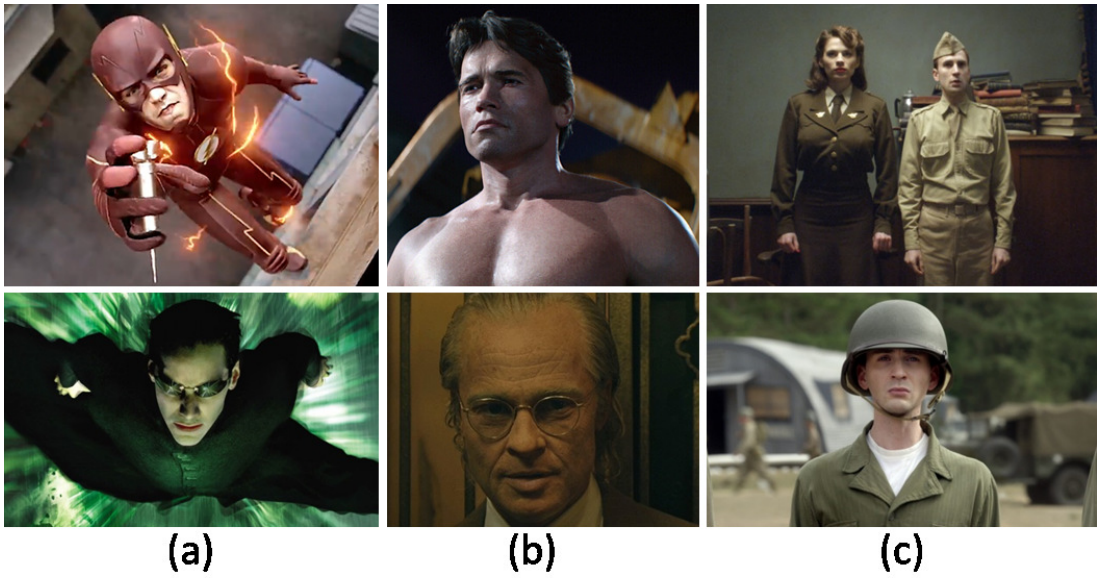


Figure 1.1: Examples showing the use of photo-realistic virtual human faces in feature films. (a) Complex action live shots, such as running up walls of a building and flying. *Top: The Flash*, http://www.cwtv.com/shows/the-flash. *Bottom: The Matrix Reloaded*, http://www.warnerbros.com/matrix; (b) Younger and older digital doubles. *Top: Terminator Genisys*, http://www.terminatormovie.com/. *Bottom: The Curious Case Of Benjamin Button*, http://www.benjaminbutton.com/; (c) Changes in facial shape, such as making the actor skinnier. *Top, bottom: Captain America: The First Avenger*, http://marvel.com/captainamerica.

Recently, state-of-the-art lightweight approaches have tried to simplify the capture step by employing commodity sensors, e. g., RGB-D cameras [Bouaziz et al. 2013; Li et al. 2013b; Thies et al. 2015] or webcams [Cao et al. 2014a; Thies et al. 2016]. However, the reconstructed 3D models lack either fine-scale details, photo-realistic albedo, or both, which are essential elements to produce compelling facial animations. As a result, based on these methods one cannot perform complex video editing tasks, such as photo-realistic face appearance and expression modification, facial reenactment, among others. Automatic digitization of photo-realistic virtual faces from standard 2D video footage then remains as an open scientific challenge to the research community.

In this thesis, we address limitations concerning the capture of faces and the manually demanding work needed in the digitization pipeline by developing novel automatic techniques that advance the state of the art in photo-realistic face capture, animation, and editing from standard monocular video recordings. More precisely, we propose robust and fully automatic methods that aim to 1) reconstruct highly-detailed fully-controllable 3D face models from monocular 2D videos and 2) photo-realistically modify faces with these models in 2D videos recorded under uncontrolled scene and illumination conditions. As a proof of concept, we test our methods on different real-life application scenarios, including face reenactment, visual dubbing, face modification, and video rewriting.

Figure 1.2: Challenges in monocular face capture and video-based editing. (a) Extreme out-of-plane head rotations. (b) Lighting changes (top) and shadows (bottom) over the face. (c) Occlusions of external objects, e. g., scalp hair and glasses (top) and non-skin features, e. g., facial hair (bottom). (d) Disocclusions in the lip region. The outer and especially the inner boundary of the lips recurrently appears (top) and disappears (bottom) during speech and as a result of complex motions.

1.2 Scope and Overview

Given an unscripted monocular 2D video of an actor recorded under unknown scene lighting, the goal of this thesis is to develop robust, accurate, and fully-automatic model-based methods for capturing high-quality facial performances, animating controllable 3D facial models at high fidelity, and editing photo-realistic faces with a plausible mouth interior.

The goal stated above is ambitious since the input video data lack 3D information and present several challenges, e. g., out-of-plane head rotations, varying illumination, (self) occlusions, and sudden and expressive facial motion, as illustrated in Figure 1.2. To simplify the problem at hand, this thesis assumes that no strong (cast) shadows and occlusions cover the face surface that we wish to reconstruct, animate, and edit. To cope with the other inherent challenges in the capture and editing of faces, we propose novel algorithms that all unify in a common framework and gradually improve on the underlying representation of the face to handle more complex video input. We start with a simple non-parametric 2D shape representation based on accurate 2D landmarks, which is then extended to a fully-controllable parametric 3D face model with multiple levels of details. This 3D model not only allows us to capture photo-realistic appearance and detailed 3D geometry in challenging unconstrained videos, but also to perform advanced photo-realistic video editing tasks with minimal user-interaction, e. g., by just modifying high-level controllers with which digital artists are familiar.

The specific technical contributions of this thesis differ in what part of the face digitization pipeline they improve. In particular, we contribute to face capture, facial animation, and face editing. First, contributions in face capture include accurate tracking of 2D facial landmarks, model-based reconstruction of detailed dynamic 3D geometry, multilayer-based reconstruction of accurate parametric 3D models, and regression-based 3D lip shape enhancement from high-quality data. Second, advances in facial animation are video-based face reenactment based on robust motion and appearance

metrics as well as temporal clustering, performance-driven retargeting of detailed facial models in sync with audio, and the automatic creation of personalized controllable 3D rigs. Finally, advances in plausible photo-realistic editing include dense face albedo capture and mouth interior synthesis using 2D image warping and 3D teeth proxies.

The contributions described in this thesis are structured according to the improvements on the face representation used internally by our methods rather than categorized by the advances in individual application areas. This organization emphasizes better the contributions for two main reasons. On the one hand, it illustrates the capabilities of the novel algorithms and models proposed in our framework for different application scenarios. On the other hand, it shows the improvements that are necessary to enable unconstrained capture as well as more sophisticated animation and editing tasks.

1.3 Structure

This thesis is divided into nine chapters from which Chapters 4–9 cover the main technical contributions in the areas of face tracking, facial animation, and face editing:

- Chapter 1 introduces the topic of this thesis, states the goals, outlines the structure of exposition, summarizes the technical chapters, and stresses the main technical contributions.
- Chapter 2 describes both the fundamental concepts and the mathematical notation that is used throughout this thesis. These are mainly concerned with face modeling, as well as the representation and synthesis of the face in the image.
- Chapter 3 provides a comprehensive overview of the related work in the following areas: Facial performance capture, lip tracking, face rig and detail generation, speech- and video-driven facial animation, and face replacement and modification in monocular videos.
- Chapters 4–9 present the main technical contributions. As mentioned before, these chapters are structured to emphasize improvements on the underlying representation of the face: From a simple non-parametric 2D shape model to a detailed and fully parametrized 3D model that allows for more robust face reconstruction in uncontrolled 2D video footage, and for realistic facial animation and video editing. Improvements on the face representation are discussed at the end of each chapter and linked to subsequent chapters in this thesis. Furthermore, each chapter shows challenging application scenarios that demonstrate the contributions in the three areas mentioned above.
- Chapter 10 summarizes the core contributions and results achieved thus far, and it briefly discusses already existing extensions as well as future challenges not explored in this thesis. Furthermore, it gives an outlook towards the full digitization of human head avatars.

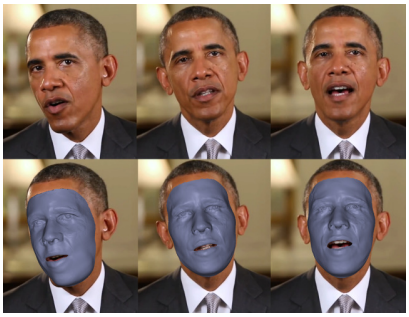
The following section gives a more detailed overview of the technical chapters of this thesis.

1.3.1 Summary of Technical Chapters

Chapter 4 introduces an automatic image-based, facial reenactment method that tracks and replaces the face of an actor in a target video with that of a user from a source video, while preserving the original target performance (published as [Garrido et al. 2014]; partially as [Garrido et al. 2013]). This method combines new image retrieval and image-based facial transfer techniques, the latter relying on accurate 2D face tracking. Compared to related approaches, the proposed method is fully automatic and robust under moderate head motion. Moreover, it does not require a tailored database of source expressions, but only short source videos with arbitrary facial motion. A user study and quantitative validations show that the proposed method generates plausible reenactments, both for self-recorded videos and for low-quality internet footage.

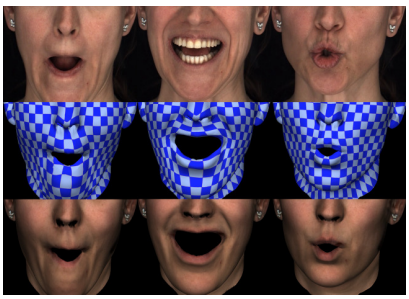
Motivated by the inability of the previous image-based approach to track and transfer challenging facial motion, Chapter 5 presents the first model-based approach for capturing detailed, dynamic, and spatio-temporally coherent 3D face geometry from markerless 2D videos (published as [Garrido et al. 2013]). This method relies on several algorithmic contributions that are non-trivially joined with state-of-the-art 2D and 3D vision and graphics techniques adapted to monocular video. Even though the proposed method requires the camera's intrinsics and a manually initialized coarse 3D model of an actor, the capturing process is fully automatic, works under fully uncontrolled lighting, and successfully reconstructs transient fine-scale skin details, e. g., wrinkles. High-quality performance capture results are demonstrated on long and expressive sequences recorded indoors and outdoors, and the relevance of the proposed approach is illustrated as an enabling technology for model-based editing of facial textures in video.

Next, Chapter 6 shows the potential of the previous model-based approach for retargeting tasks in real-life applications, namely dubbing in movies (published as [Garrido et al. 2015]). More specifically, it presents the first approach that alters the mouth motion of a target actor in a video, so that it matches a new audio track spoken in a different language by a dubbing actor. This approach builds upon monocular performance capture and scene lighting estimation (see Chapter 5). It also exploits audio analysis in combination with space-time frame retrieval to render new photo-realistic 3D shape models of the mouth region to replace the original target performance. A user study and qualitative validations show that the proposed approach produces plausible results on par with footage that has been professionally dubbed in the traditional way.



Chapter 7 introduces a novel multilayer model-based approach for capturing arbitrary 3D face performances from 2D videos with unknown camera, scene and lighting setups (published as [Garrido et al. 2016a]). The heart of this approach is a new multilayer parametric face model that jointly encodes plausible facial appearance and 3D geometry variation that is represented at multiple layers of detail. The appearance is modeled by the incident lighting and an estimate of the face albedo, while the shape is encoded by a subspace of facial shape identity, facial expressions, person-specific medium-scale corrective shapes, and fine-scale skin details. These layers are optimized automatically in a new inverse rendering framework that exploits color cues and accurate 2D landmark trajectories. The proposed method is tested on challenging unconstrained sequences, e. g., YouTube videos. Qualitative and quantitative experiments confirm that this novel multilayer approach produces results of higher quality than the approach from Chapter 5 and competes with or even outperforms other state-of-the-art approaches.

Next, Chapter 8 presents an automatic approach to the creation of high-quality, personalized 3D face rigs that can be intuitively controlled by high-level expression controllers (also published as [Garrido et al. 2016a]). These face rigs are based on three distinct layers (coarse, medium and fine) and learned using a novel sparse regression approach. The proposed regression approach couples the coarse layer represented as generic expressions (i. e., blendshapes) to the medium and fine-scale layers, each containing different levels of personalized shape details. Such a coupling assures local semantic control of personalized deformations in ways consistent with expression changes. Different application scenarios demonstrate that the reconstructed face rigs when combined with the estimated scene lighting and personalized skin albedo open up a world of possibilities in realistic facial animation and for more complex video editing tasks.



Finally, Chapter 9 addresses the problem of accurate capture of 3D lip shapes. It presents a fully automatic data-driven approach to reconstruct detailed and expressive lip shapes, along with the dense geometry of the face, from a monocular video (published as [Garrido et al. 2016b]). At its core is a new gradient-domain lip correction network that leverages 2D lip contours and coarse 3D lip geometry to learn the difference between inaccurate and ground-truth 3D shapes of lips, where ground truth lip shapes are obtained from a new database of high-quality multiview reconstructions. Quantitative and qualitative results demonstrate that the proposed method improves the reconstruction of complex lip motions when compared to state-of-the-art monocular tracking, and it also generalizes well to general scenes and unseen individuals.

1.4 Technical Contributions

In the following, we provide a more detailed list of technical contributions that enable the methods described above.

The main contributions of Chapter 4 are:

- Accurate localization of a sparse set of 2D landmarks based on optical flow correction between automatically selected keyframes.
- A novel distance metric, which combines both appearance and motion information, to retrieve similar facial expressions between videos, while preserving temporal continuity.
- A new temporal clustering that groups similar target expressions into consecutive clusters to stabilize matching and assure accurate image selection.
- A simple, yet robust, image-based warping strategy that preserves the actor's face shape (i. e., identity), while providing sufficiently precise head motion.

The main contributions of Chapter 5 are summarized as follows:

- Automatic, drift-free model-based tracking, which succeeds on long sequences with expressive faces and fast motion, based on a sparse set of accurate 2D landmark trajectories.
- Temporally-coherent dense 3D geometry correction through a novel multi-frame variational optical flow approach.

The main contributions of Chapter 6 are:

- A performance capture-based system for video-realistic retargeting and resynthesis of detailed performances that align the visual channel with a dubbed audio signal.
- A spatio-temporal rearrangement strategy that uses the input facial performances and the dubbed audio channel to synthesize new highly-detailed and synchronized 3D target performances.
- Reconstruction of realistic target face albedo and synthesis of a plausible mouth interior based on a geometric teeth proxy and 2D image warping.

The main contributions of Chapter 7 are outlined as follows:

- A new parametric facial shape representation to reconstruct and represent the 3D facial surface at different levels of detail.
- A unified novel fitting approach that leverages both color cues and a sparse set of accurate 2D landmarks to reconstruct coarse- and medium-scale facial shape.

The main contributions of Chapter 8 are:

- Automatic extraction of parametrized rigs that model the correlation between blendshape weights and person-specific idiosyncrasies at a medium- and a fine-scale detail layer.

- A novel sparse regression approach that exploits the local support of blendshapes to produce more accurate, detailed and realistic face rig animations.

The main contributions of Chapter 9 are summarized as follows:

- A novel, high-quality 3D lip shape database containing challenging motions, such as rolling and extreme lip deformations, and general speech animations.
- A new data-driven strategy that learns accurate 3D lip deformations from high-quality multi-view reconstructions enhanced with lip marker data.
- A robust gradient domain regression algorithm trained to infer accurate lip shapes from sub-optimal monocular reconstructions and automatically detected 2D lip contours.

1.5 List of Publications

The work presented in this thesis mainly encompasses five peer-reviewed scientific publications, published at top-tier conferences and journals in the field of computer graphics and vision. These papers address challenging problems in facial performance capture and face capture-based animation and editing from monocular video. In addition, this thesis briefly discusses in Chapter 10 a co-authored paper that goes beyond face digitization and reconstructs detailed, personalized 3D teeth models in non-invasive capture setups.

The five papers in the area of face capture, animation and editing are:

- P. Garrido, L. Valgaerts, C. Wu and C. Theobalt. "Reconstructing detailed dynamic face geometry from monocular video". *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 32(6), 158:1-158:10, 2013.
- P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormaehlen, P. Pérez and C. Theobalt. "Automatic face reenactment". In *CVPR*, 4217-4224, IEEE, 2014.
- P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Pérez and C. Theobalt. "Dub: Modifying face video of actors for plausible visual alignment to a dubbed audio track". *Comput. Graph. Forum (Proc. Eurographics)*, 34(2), 193-204, 2015.
- P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. "Reconstruction of personalized 3D face rigs from monocular video". *ACM Trans. Graph.*, 35(3), 28:1–28:15, 2016a.
- P. Garrido, M. Zollhöfer, C. Wu, D. Bradley, P. Pérez, T. Beeler and C. Theobalt. "Corrective 3D reconstruction of lips from monocular video". *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 35(2), 219:1–219:11, 2016b.

The co-authored paper that addresses the problem of teeth and gum reconstruction from images and video is:

- C. Wu, D. Bradley, P. Garrido, M. Zollhöfer, C. Theobalt, M. Gross and T. Beeler. "Model-based teeth reconstruction". *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 35(6), 220:1–220:13, 2016.

Chapter 2

Basics

2.1 Facial Animation and Modeling

2.1.1 Blendshapes

Blendshapes are extensively used by animation artists in 3D modeling and animation due to their underlying semantic meaning. They can be thought of as additive facial expressions built on top of a neutral face (see Figure 2.1). Mathematically, they form an additive model of potentially non-orthogonal linear deformations and, in principle, any new facial expression can be approximated by a weighted (or convex) combination of blendshapes [Lewis et al. 2014]. Let $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ be the set of n blendshapes and $\mathbf{b}_0$ be the neutral face, where $\mathbf{b}_i \in \mathbb{R}^{3k}, \forall i$ are represented as column vectors, and k denotes the total number of vertices depicting the 3D face shape. A new facial expression $\mathbf{e}$ can then be obtained as a linear combination of blendshapes[1], yielding the so-called blendshape model:

$$\mathbf{e} = B\mathbf{a} = \sum_{i=0}^{n} \alpha_i \mathbf{b}_i \ , \tag{2.1}$$

where $0 \leq \alpha_i \leq 1$, $\forall i = 0 : n$ denote the linear weights (oftentimes controlled by sliders), $\mathbf{a} = [\alpha_0, \ldots, \alpha_n]^\top \in \mathbb{R}^{n+1}$ and $B = [\mathbf{b}_0 \,|\, \mathbf{b}_1 \,|\, \mathbf{b}_2 \,|\, \cdots \,|\, \mathbf{b}_n] \in \mathbb{R}^{3k \times (n+1)}$ is the basis of variation in expression, represented as a stack of blendshapes (including the neutral face).

The formulation in Equation 2.1 imposes an undesired global scaling factor when combining different blendshapes. This is normally counteracted by imposing hard constraints on the sum of weights, i. e., $\sum_i \alpha_i = 1$. A more convenient and popular representation used by many modeling packages (e. g., Maya) and different approaches in the literature [Bouaziz et al. 2013; Li et al. 2010; Li et al. 2013b; Thies et al. 2015; Weise et al. 2011] is to model the blendshapes as delta variations that linearly add up on top of the neutral face:

$$\mathbf{e} = \mathbf{b}_0 + B\mathbf{a} = \mathbf{b}_0 + \sum_{i=1}^{n} \alpha_i (\mathbf{b}_i - \mathbf{b}_0) = \mathbf{b}_0 + \sum_{i=1}^{n} \alpha_i \mathbf{d}_i \ , \tag{2.2}$$

[1] Note that a solution is feasible if and only if the new facial expression can be obtained by interpolation, i. e., if the model can explain such an expression by a linear combination.

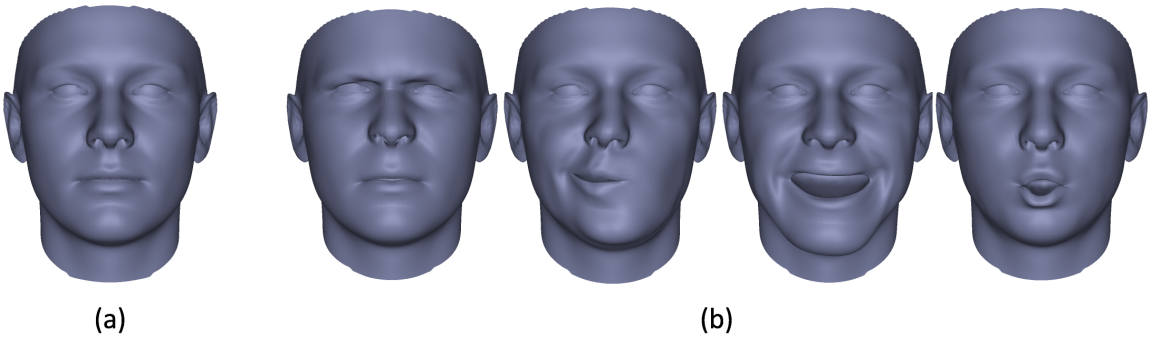


Figure 2.1: Example of a blendshape model. (a) Neutral face. (b) Semantic shapes. *From left to right*: Disgust, mouth to the right, smile, and funneler (i. e., "O"-like mouth shape).

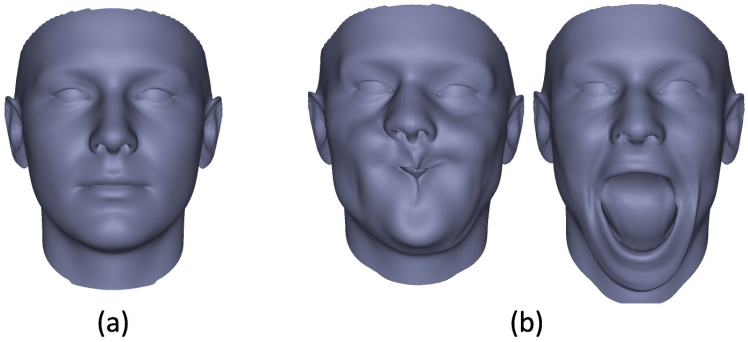


Figure 2.2: Artifacts produced by the linear dependency between the blendshapes. (a) Neutral face. (b) Shape artifacts. *Left:* Shape inconsistency due to activation of left and right mouth motion. *Right:* Unrealistic mouth shape due to activation of similar shapes (wide open smile + mouth open).

where $0 \leq \alpha_i \leq 1, \forall i = 1:n$, $\mathbf{a} = [\alpha_1, \ldots, \alpha_n]^\top \in \mathbb{R}^n$ and $B = [\mathbf{d}_1 \mid \mathbf{d}_2 \mid \cdots \mid \mathbf{d}_n] \in \mathbb{R}^{3k \times n}$ is the basis of variation in expression, represented as per-vertex 3D displacements. In this thesis, we will employ this delta formulation unless stated otherwise.

Although the box constraints imposed on the linear weights $\alpha_i, \forall i$ control the influence of blendshapes in the model ($\alpha = 0$ deactivated blendshape; $\alpha = 1$ fully-activated blendshape), some blendshapes simply cannot be combined together due to shape inconsistencies caused by the linear dependency of the vectors. For instance, due to anatomical face symmetry constraints, moving the mouth to the left and to the right at the same time is not allowed and leads to distortions (see Figure 2.2). Analogously, the combination of semantically similar expressions, e. g., a wide open smile combined with a mouth open, adds a double effect and may normally result in unrealistic deformations (see Figure 2.2). This problem can be alleviated by utilizing pairwise activation constraints of the form $\alpha_i \alpha_j = 0, \forall i \neq j$ [Lewis et al. 2014], or by employing a strong prior that enforces sparsity [Bouaziz et al. 2013] or restricts the activation of linear weights [Li et al. 2013b; Thies et al. 2015]; however, this does not completely prevent inconsistent blendshape combinations. Despite these limitations, blendshape models are normally preferred over principal component analysis (PCA) models as they provide a more intuitive control of facial expressions with meaningful parameter dimensions. As such, blendshapes are widely utilized by animation artists to perform different retargeting or animation-related tasks.

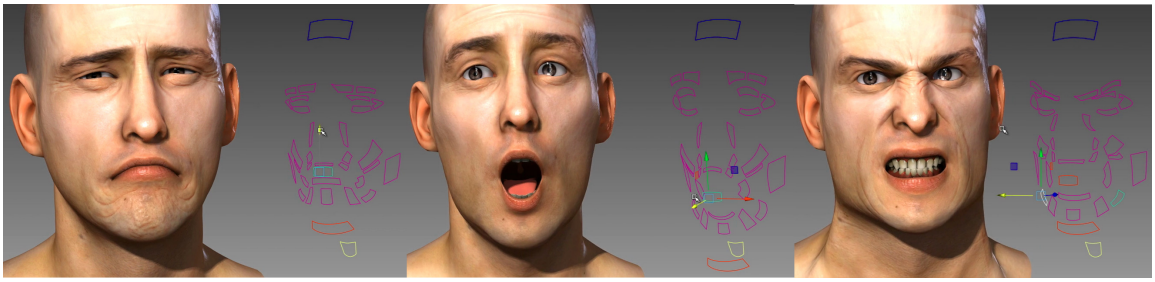


Figure 2.3: Photo-realistic, personalized 3D face rig that has been created manually by an artist, https://vimeo.com/soukizero. The 3D rig is driven by custom-made blendshape controllers (shown next to the rig).

2.1.2 Facial Rig

In animation, the term rig normally refers to a bone structure attached to the muscles and skin of a digital character. Such a structure allows digital artists to have full control over the character's coarse motion and dynamics, while still reproducing realistic surface deformations – a crucial task in animation. In facial animation, however, rigs are normally not conceived as rigid structures, but more general deformable surfaces that control not only rigid deformation of the jaw or eyes, but also expressions, skin stretching, muscle bulging, and lip motion, among others. As such, face rigs represent the face dynamics and character-specific idiosyncrasies (e. g., personalized smiles and frown lines) that are necessary to create believable facial animations of an actor. Face rigs can be created using either detailed tailor-made blendshapes, physically-based geometric deformations driven by simulated muscle activations, or a combination of both [Komorowski et al. 2010]. These rigs are then dynamically controlled or animated by artists using high-level controllers that steer person-specific facial deformations.

Photo-realistic face rigs, which are of major interest for this thesis, often require hundreds of custom controllers or handlers to model actor-specific facial expressions, face appearance, and soft tissue deformation, such as wrinkles and folds (see Figure 2.3). To create convincing photo-realistic rigs that do not fall in the uncanny valley, digital artists normally require high-quality 3D scans of an actor (neutral face plus some standard key expressions) captured in complex multiview camera systems [Klehm et al. 2015]. Yet, the sculpting of complex facial details and face dynamics as well as the rigging process is an artistic manual work that may take several weeks (if not months) before completion. In Chapter 8, we propose the first approach that automatically generates a highly-detailed facial rig from unconstrained monocular video data. The reconstructed rig can be controlled with intuitive blendshape sliders and can be used as high-quality prototypes to sketch facial animations without going through the entire conventional digitization pipeline in post-production, thus saving time and manual effort.

2.2 Camera and Image Formation Model

2.2.1 Camera Model

To represent a 3D object in the scene and its corresponding 2D projection onto the image plane, we assume a simple camera model, where a 2D image point $\mathbf{p}$ is formed by projecting a 3D world point

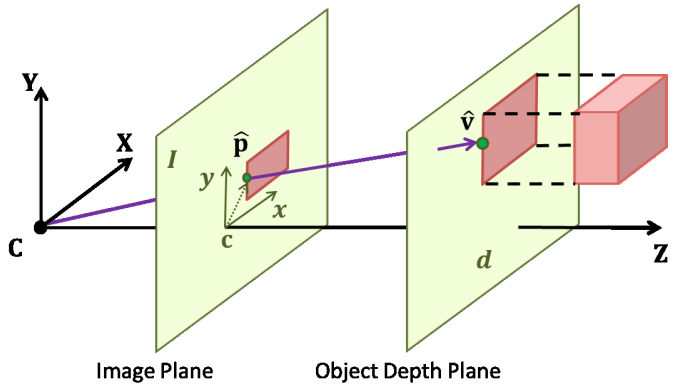


Figure 2.4: Weak perspective camera model. Objects undergo a two-step projection. First, the object's geometry is flattened in depth using an orthographic projection (optical rays are parallel). Second, the flattened geometry is globally rescaled based on its distance to the camera. Here, the image coordinate system is spanned by the vectors x and y, and the camera's intrinsics are given by the principal point $\mathbf{c} = [\mathbf{c}_x, \mathbf{c}_y]^\top$.

$\mathbf{v}$ using a perspective transformation [Forsyth and Ponce 2012], as follows:

$$\mathbf{p}(K, R, \mathbf{t}) = K\Pi(R\mathbf{v} + \mathbf{t}) = K\Pi(\hat{\mathbf{v}}) \ , \tag{2.3}$$

where $[R|\mathbf{t}] \in \mathbb{R}^{3\times4}$ refers to the camera's rigid transformation (also called camera extrinsics) that transforms the 3D point $\mathbf{v}$ into a point $\hat{\mathbf{v}}$, represented in camera coordinates. Here, $\Pi(\cdot)$ denotes a (non-)linear operator that projects the aligned 3D point $\hat{\mathbf{v}}$ onto the 2D image plane, and K is the geometric property of the camera, also known as camera intrinsics. Note that $\mathbf{p} = [\mathbf{p}_x, \mathbf{p}_y, 1]^\top$ is the projection of $\mathbf{v}$ onto the image plane in homogeneous coordinates. In non-homogeneous screen space, this point is represented as $\hat{\mathbf{p}} = [\hat{\mathbf{p}}_x, \hat{\mathbf{p}}_y]^\top$.

Weak Perspective Camera Model

The weak perspective model is a simplified, yet reasonable, model commonly used in computer vision, since it represents the projection of an object onto the image plane as a simple linear operator. In this model, optical rays are assumed to be orthogonal to the camera plane up to a scaling factor (see Figure 2.4), yielding the following projection operator in homogeneous coordinates:

$$\Pi(\cdot) \quad = \quad \rho \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \ , \tag{2.4}$$

where $\rho = 1/d$ is the scaling factor that accounts for global changes in depth d (i. e., proximity of objects to the camera plane). Thus, an object is considered as a plane that virtually appears bigger or smaller in the projection depending on its distance to the camera.

To represent a pixel $\mathbf{p}$ in homogeneous image coordinates, the matrix of intrinsics parameters K is defined as follows:

$$K \quad = \quad \begin{pmatrix} 1 & 0 & \mathbf{c}_x \\ 0 & 1 & \mathbf{c}_y \\ 0 & 0 & 1 \end{pmatrix} \ , \tag{2.5}$$

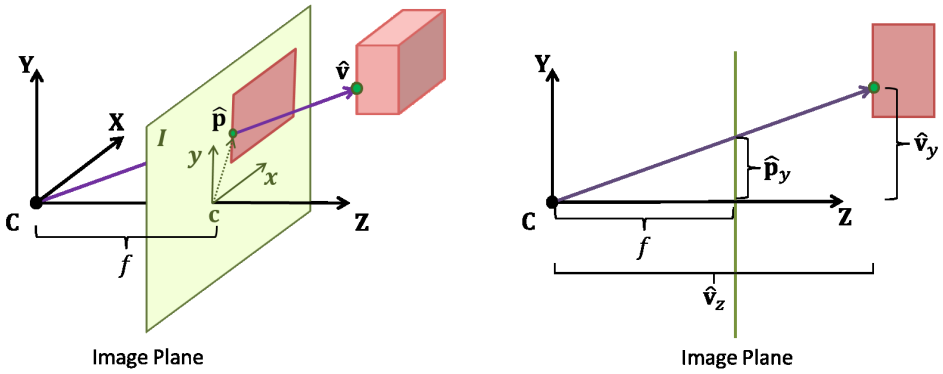


Figure 2.5: Full perspective camera model. A 3D point $\hat{\mathbf{v}}$ is projected onto the image plane at position $\hat{\mathbf{p}}$ using non-linear perspective projection. Here, the image coordinate system is spanned by the vectors x and y, and the intrinsic properties are given by the focal length f and the principal point $\mathbf{c} = [\mathbf{c}_x, \mathbf{c}_y]^\top$.

where $\mathbf{c} = [\mathbf{c}_x, \mathbf{c}_y]^\top$ is called the principal point and represents the intersection between the optical axis and the image plane of the camera. In this thesis, $\mathbf{c}$ lies at the image origin unless stated otherwise. It is important to remark that the weak perspective camera model will be partly employed in Chapters 5–6.

Full Perspective Camera Model

Here, the projection of an object onto the image plane is represented by a full perspective camera model (often referred to as a pinhole camera model) where optical rays converge at the image center. The projective geometry in the camera sensor is mainly determined by the focal length f and the principal point $\mathbf{c} = [\mathbf{c}_x, \mathbf{c}_y]^\top$ (see Figure 2.5). For the sake of simplicity, let us first assume that the principal point lies at $\mathbf{c} = [0,0]^\top$. By using similarity of triangles, we can associate a 3D point $\hat{\mathbf{v}}$ with a pixel $\hat{\mathbf{p}}$ in the sensor optics as follows:

$$\frac{\hat{\mathbf{p}}_x}{\hat{\mathbf{v}}_x} = \frac{\hat{\mathbf{p}}_y}{\hat{\mathbf{v}}_y} = \frac{f}{\hat{\mathbf{v}}_z} \quad . \tag{2.6}$$

In the sensor optics, $\hat{\mathbf{v}}$ undergoes a non-linear perspective projection up to a factor given by the focal length f:

$$\hat{\mathbf{p}}_x = f\frac{\hat{\mathbf{v}}_x}{\hat{\mathbf{v}}_z}, \;\; \hat{\mathbf{p}}_y = f\frac{\hat{\mathbf{v}}_y}{\hat{\mathbf{v}}_z} \quad . \tag{2.7}$$

If we represent this transformation in homogeneous coordinates, the point $\hat{\mathbf{v}}$ is first projected using the non-linear operator $\Pi(\hat{\mathbf{v}}) = [\hat{\mathbf{v}}_x/\hat{\mathbf{v}}_z, \hat{\mathbf{v}}_y/\hat{\mathbf{v}}_z, 1]^\top$. Then, to properly represent a 2D point $\mathbf{p}$ in the camera plane under an arbitrary optical center, the matrix of intrinsics parameters K is defined as follows:

$$K \;\; = \;\; \begin{pmatrix} f & 0 & \mathbf{c}_x \\ 0 & f & \mathbf{c}_y \\ 0 & 0 & 1 \end{pmatrix} \quad , \tag{2.8}$$

As stated above, $\mathbf{c}$ lies at the image center unless stated otherwise. The focal length f can be calibrated beforehand [Bradski and Kaehler 2013; Zhang 2000] or estimated while tracking the face

(see Chapter 7). Note that the full perspective camera model will be partly used in Chapters 5–6, but then fully adopted in the latest chapters (see Chapters 7–9).

2.2.2 Image Formation Model

Even though complex light transport mechanisms such as subsurface scattering exist, we assume a pure Lambertian reflection model to represent the incident lighting on the face surface, i. e., an isotropic diffuse BRDF that reflects radiance equally into all directions. This assumption has been commonly used in the literature [Wu et al. 2011b; Valgaerts et al. 2012b; Garrido et al. 2013; Thies et al. 2015; Garrido et al. 2016a]. Note that we reckon the face in the scene as a non-emitter.

Let us now define $L(\hat{\mathbf{v}}, \omega) \in \mathbb{R}^3$ as the incident lighting at a mesh vertex $\hat{\mathbf{v}}$ from an incoming light direction $\omega \in \mathbb{R}^3$. Note that $L(\hat{\mathbf{v}}, \omega)$ is represented as RGB illumination, i. e., non-white illumination. The rendering equation can be then defined as follows:

$$\mathcal{B}(\hat{\mathbf{v}}, \omega) = \mathbf{c}(\hat{\mathbf{v}}) \circ \int_{\Omega} L(\hat{\mathbf{v}}, \omega) V(\hat{\mathbf{v}}) \max(\langle \omega, \hat{\mathbf{n}}(\hat{\mathbf{v}}) \rangle, 0) \, \mathrm{d}\omega \ , \tag{2.9}$$

where $\mathcal{B}(\hat{\mathbf{v}}, \omega)$ is the irradiance at vertex $\hat{\mathbf{v}}$ from direction ω sampled on the hemisphere Ω, $\mathbf{c}(\hat{\mathbf{v}}) \in \mathbb{R}^3$ denotes the skin albedo at vertex $\hat{\mathbf{v}}$, $\hat{\mathbf{n}} \in \mathbb{R}^3$ represents the normal at vertex $\hat{\mathbf{v}}$, and $V \in \{0, 1\}$ is a binary function that measures the visibility of point $\hat{\mathbf{v}}$ w. r. t. the camera view point, which is assumed to be known. Here, $\langle \cdot \rangle$ represents the inner product and $\circ$ denotes a point-wise multiplication.

Let us redefine w. l. o. g. $L(\omega)$ as the incident lighting at vertex $\hat{\mathbf{v}}$. In this thesis, we approximate the lighting $L(\omega)$ using spherical harmonics (SH) functions as in [Wu et al. 2011b; Valgaerts et al. 2012b], yielding the following formula:

$$L(\omega, \Gamma) = \sum_{l=0}^{j-1} \sum_{m=-l}^{l} \gamma_l^m Y_l^m(\omega) \ , \tag{2.10}$$

where $Y_l^m \in \mathbb{R}$, $\forall l, m$ denote the SH functions, $\Gamma = [\gamma_0^0, \gamma_1^{-1}, \gamma_1^0, \gamma_1^1, \dots]$ are the coefficients of the SH basis, j is the number of bands, and l is the index of the band. Here, $\gamma = [\gamma^r, \gamma^g, \gamma^b]^\top$ is a three valued-vector that increases or decreases the effect of the lighting at each channel. We remark that in this work we use $j = 4$ bands unless stated otherwise. For the sake of simplicity, we can re-write Equation 2.10 in a more compact form, as follows:

$$L(\omega, \Gamma) = \sum_{l=1}^{j^2} \gamma_l Y_l(\omega) \ . \tag{2.11}$$

By inserting Equation 2.11 into Equation 2.9, we obtain:

$$\mathcal{B}(\hat{\mathbf{v}}, \omega, \Gamma) = \mathbf{c}(\hat{\mathbf{v}}) \circ \int_{\Omega} \sum_{l=1}^{j^2} \gamma_l Y_l(\omega) V(\hat{\mathbf{v}}) \max(\langle \omega, \hat{\mathbf{n}}(\hat{\mathbf{v}}) \rangle, 0) \, \mathrm{d}\omega \ . \tag{2.12}$$

Instead of sampling all over the hemisphere Ω every time the face surface changes, we can sample D incoming directions ω around a unit sphere (for instance, using Hammersley sampling) and keep them fixed, yielding a coarse and discrete representation of the incident lighting that can be precomputed in advance. As a result, we obtain a discrete approximation of the rendering equation that is no longer parametrized in terms of ω:

$$\mathcal{B}(\hat{\mathbf{v}}, \Gamma) \quad = \quad \mathbf{c}(\hat{\mathbf{v}}) \circ \frac{4\pi}{D} \sum_{d=1}^{D} \sum_{l=1}^{j^2} \gamma_l Y_l(\omega_d) V(\hat{\mathbf{v}}) \max(\langle \omega_d, \hat{\mathbf{n}}(\hat{\mathbf{v}}) \rangle, 0) \ . \tag{2.13}$$

However, this approximation heavily depends on the quality of the sampling strategy and the number of sample locations, making it both inefficient and inaccurate. To overcome this limitation, recent methods consider the outgoing lighting reflected by the surface, which can be approximated by the surface normals [Wu et al. 2013; Thies et al. 2015]. This leads to the following formula:

$$\begin{aligned} \mathcal{B}(\hat{\mathbf{v}},\Gamma) &= \mathbf{c}(\hat{\mathbf{v}}) \circ 4\pi \sum_{l=1}^{j^2} \gamma_l Y_l(\hat{\mathbf{n}}(\hat{\mathbf{v}}))V(\hat{\mathbf{v}}) \\ &= \mathbf{c}(\hat{\mathbf{v}}) \circ 4\pi \hat{L}(\hat{\mathbf{v}},\Gamma) \ . \end{aligned} \tag{2.14}$$

This approximation allows for a faster and more robust estimation of the illumination that does not depend on the quality of the sampling. In this thesis, we employ this approximation to model the reflectance of the face surface, unless explicitly stated otherwise.

Let us now assume that the irradiance of the object is known (i. e., the coefficients of the SH functions have been already estimated) and define $\mathcal{B}(\hat{\mathbf{v}})$ as the color assigned to the corresponding vertex $\hat{\mathbf{v}}$. To render the object in the image, each vertex $\hat{\mathbf{v}}$ is projected onto the image plane at position $\hat{\mathbf{p}}$ using the camera model described in Section 2.2.1. Finally, the color of pixel $\hat{\mathbf{p}}$ is assigned to $\mathcal{B}(\hat{\mathbf{v}})$ via direct lookup.

Chapter 3

Related Work

This chapter provides a survey of the three most important topics covered in this thesis, namely face capture and tracking, facial animation, and editing of faces in 2D video sequences. More precisely, it reviews the related work on facial performance capture (Section 3.1), lip tracking (Section 3.2), face rig and detail generation (Section 3.3), speech- and video-driven facial animation (Section 3.4), as well as face replacement and modification in monocular videos (Section 3.5).

3.1 Facial Performance Capture

Facial performance capture techniques commonly aim to reconstruct robust and accurate facial motion/expressions, highly-detailed dynamic facial models (either 2D or 3D models), and possibly the appearance of the face from optical-based sensor measurements of an actor's performance. Such reconstructions can potentially enable us to animate realistic avatars that accurately mimic the actor's expressions or generate photo-realistic digital characters for movies, provided that the mannerisms, as well as the facial details and texture of the actor's face, are accurately acquired. Thus, facial performance capture is a crucial step for believable facial animation.

Researchers in the area have tried to achieve this goal using sophisticated indoor capture systems that are expensive to build, but recently there has been an interest to push the frontier even further by capturing performances from low-cost devices, such as RGB-D sensors or even ubiquitous monocular cameras, as in this thesis work.

This section gives a survey of different methods that attempt to solve this challenging problem.

3.1.1 Dense Facial Performance Capture

Most algorithms for dense (and often very detailed) 3D facial performance capture resort to motion capture data, structured light systems, or complex and dense camera arrays that may even rely on sophisticated lighting patterns to track 3D surface geometry [Pighin and Lewis 2006; Klehm et al. 2015]. Note that this section gives only a brief review of the main methods, since capturing 3D facial models from a single camera is the primary focus of this thesis.

Marker-based Motion Capture

In this category, we find methods that typically use dense camera sets and markers (or also invisible makeup) to track and deform an existing 3D template of the actor's face.

The basic idea dates back to Williams et al. [1990] where a 3D surface geometry of an actor's face with neutral expression and fixed texture is deformed using sparse 2D motion capture (mocap) data from video. New smooth expressions are generated by employing interpolation kernels distributed over the markers in the 3D geometry. Guenter et al. [1998] used a more sophisticated system that renders expressive 3D faces by leveraging denser 3D mocap data (around 200 dots) and dynamic texture maps. Dense 3D markers are retrieved by tracking painted dots on the actor's face from several cameras, which in turn are utilized to deform the 3D facial geometry by linear blending. Dynamic texture data can be obtained at each frame due to a consistent parametrization of the tracked face geometry.

Nowadays, Vicon greatly dominates the commercial market for cutting-edge 3D maker-based facial capture [VICON]; however, due to the low spatial resolution (about 100-200 markers), they cannot capture wrinkle patterns over the face. In [Furukawa and Ponce 2009; Bickel et al. 2007], wrinkles and folds are additionally captured by leveraging visual cues (using either visible or invisible makeup) from multiple videos. Bickel et al. [2007] proposed a multi-scale capture approach that additionally estimates medium-scale folds by inverse rendering. Folds are tracked using two synchronized cameras based on user-defined painted regions, and their shape is parametrized by 2D B-splines. The final 3D shape is synthesized by minimizing a non-linear shell energy that preserves surface area and curvature, yielding the desired bulges around regions with wrinkles. Furukawa et al. [2009] introduced an alternative approach that uses dense makeup as optical cues to accurately estimate shape deformations. To capture complex skin stretch and shear, the system explicitly models and adaptively estimates tangential non-rigid deformation, which is assumed to be piece-wise smooth over local structures. This estimate is in turn used to define a tangential rigidity term that regularizes the deformation of the 3D shape, i. e., stretching of edges, in an adaptive manner. Mova Contour facial performance capture technology [MOVA] is another commercial system that similarly resorts to dense fluorescent makeup to accurately track face geometry and reconstruct fine-scale skin details, such as folds and wrinkles.

An orthogonal approach proposed in [Huang et al. 2011] leverages highly-detailed, registered 3D facial scans to generate a minimal blendshape basis, thus reducing the capture problem to estimating the optimal blendshape combination that matches the sparse 3D mocap data. The optimal set of registered scans (i. e., blendshapes) is selected using a greedy strategy based on reconstruction errors. Here, sparse 3D correspondences between the mocap data and the set of scans are found by a rigid and non-rigid registration method based on iterative closest points (ICP). Dense 3D correspondences across facial scans are obtained by deforming a template scan to each in the set using Laplacian regularization [Sorkine et al. 2004] and optical flow constraints [Papenberg et al. 2006].

Structured Light Systems

Structured light techniques commonly track shape templates from dynamic 3D scanner data in realtime by combining monocular or stereo video and active illumination.

In [Zhang et al. 2004], a spacetime stereo approach was proposed to capture detailed geometry, texture, and motion. Here, globally consistent dynamic depth maps are obtained by generalizing the stereo matching problem to spatio-temporal oriented windows, optimized to small blocks of

data for scalability. Textured facial models that preserve correspondences are then built by fitting a template mesh to the depth maps while enforcing optical flow constraints. They showed that new consistent animations could be created interactively by simply blending nearby meshes, using either user-defined control points or motion graphs.

Huang et al. [2004] used a monocular sinusoidal phase shift acquisition method that fits a multiresolution face mesh to depth maps. Global coarse deformations are obtained by ICP-based rigid alignment to the depth maps, followed by region-based deformations using a physics-based synthesis framework. Local deformations are modeled using free-form deformations in a Euclidean distance transform space based on cubic B-splines, and they are obtained by minimizing the difference to the scanner data in a least squares sense. Wang et al. [2004] employed this acquisition framework to learn a generative model that decomposes person-specific facial expressions into generic content and style. To do so, the facial expressions are projected into a non-linear manifold using local linear embedding and then normalized to establish correspondences, thereby creating a unified embedding. Generalized radial basis functions with linear weights (i. e., linear maps) are utilized to model the manifold. Finally, a generative model is learned using a bilinear model that separates linear weights (style) from non-linear functions (content). Such a model could be used for dynamic morphing and expression transfer, both very relevant tasks in facial animation.

Inspired by the acquisition setup proposed in [Huang et al. 2004], Weise et al. [2007] presented a robust stereo phase-shift method that can reconstruct depth maps of complex deformable objects in realtime by harnessing data parallelization on the GPU. Discontinuities and motion artifacts that may appear during the phase unwrapping of the projector data are explicitly handled by exploiting stereo data and deriving an analytical expression for the motion error incurred by the captured system. In [Weise et al. 2009], the same system is utilized for live facial puppetry. To approach this problem, they used as a prior a generic template mesh to reconstruct the actor's face and obtained consistent correspondences across his/her performed expressions using dense optical flow constraints. A person-specific parametric statistical model from these dynamic facial expressions is then created, thus simplifying the puppeteering problem to transferring source expression weights to a target face model in a linear subspace that spans the source expressions in the target space (i. e., deformation transfer space [Sumner and Popović 2004]).

Dense image-based methods

Dense image-based approaches help overcome the limitations of purely geometric and scanner-based methods, especially regarding the tracking accuracy and the quality of the surface detail. To produce high-quality facial performances, these methods typically combine mesh tracking with passive multiview stereo reconstruction obtained from complex and expensive HD camera setups.

The first passive method that requires no template mesh was proposed in [Bradley et al. 2010]. Per-frame facial geometry and texture are captured from multiview stereo data using a constrained binocular reconstruction that iteratively removes outliers. The initial reconstruction is propagated through time using optical flow constraints and then deformed with the already reconstructed meshes. Temporal drift due to extreme motions is partially corrected by analyzing flow displacements in the reference texture map, and stable mouth tracking is explicitly enforced using sparse edge-based constraints whose correspondences remain fixed in the mesh. Spatial and temporal noise is controlled with smoothing at the expense of less detailed meshes. Borshukov et al. [2003] recreated actors for *The Matrix Reloaded* using the Universal Capture system that requires laser-scanned models. As in [Bradley et al. 2010], optical flow and camera triangulation constraints allow for an

accurate tracking of the model over time, and time-varying texture maps are computed from multiple cameras. However, drift and tracking errors are corrected manually. Realistic skin scattering is simulated by capturing subtle illumination changes in a 2D lightmap and blurring it accordingly.

Beeler et al. [2011] introduced an anchor-based face tracking approach that addresses temporal drift. It is based on a high-quality single-shot reconstruction technique proposed by [Beeler et al. 2010] that captures per-frame facial scans with different topologies using a multiview binocular stereo method similar to [Bradley et al. 2010]. However, it can retrieve details at the pore level by employing a mesoscopic augmentation heuristic that uses high-pass filtered image cues, akin to shape-from-shading based refinement [Wu et al. 2011a; Valgaerts et al. 2012b]. In [Beeler et al. 2011], anchor frames are selected by analyzing cross-correlation errors between a reference image and uniformly sampled candidates. Dense correspondences to the reference image are then computed by block-based normalized cross-correlation, and correspondences to unanchored frames are propagated between anchored frames in a forward and backward direction to prevent drift. The reference mesh is then deformed using the tracked motion fields as in [Bradley et al. 2010]. Finally, the motion and shape of the deformed meshes are refined separately to assure per-frame spatial fidelity while being temporally consistent with the reference frame. A commercial system, called Dimension Imaging [DI4D] also falls into this category. As in [Beeler et al. 2011; Bradley et al. 2010], this system utilizes dense passive stereo photogrammetry and employs optical flow constraints to track a mesh with fixed topology at high fidelity.

Beeler et al. [2014] extended the system in [Beeler et al. 2011] and introduced an anatomically-inspired rigid stabilization approach to align tracked meshes. The stabilization is decomposed into two steps: Skull fitting and mesh stabilization. A generic skull is first rigidly aligned to manually annotated landmarks on the actor's neutral shape and then non-rigidly fitted using a linear shell energy that minimizes bending and stretching. Finally, new expressions are stabilized such that two main anatomical constraints are preserved: Volumetric skin constraints, and nose stretching and compression constraints. In practice, this is achieved by minimizing an energy that penalizes deviations from the predicted tissue thickness and the length of the nose on the mesh surface.

Alternatively, pore-level skin detail at millimeter precision can also be reconstructed by additionally resorting to sophisticated light stages under fully controlled illumination, and by employing custom photometric cues based on spherical-gradient illumination. In [Ma et al. 2007], static texture and detailed 3D geometry are captured using the so-called Light Stage 5, which consists of 15 binocular stereo arrays and about 150 LED lights. Here, diffuse reflectance and geometry are captured from cross polarized cameras under different spherical gradient illumination, whereas specularities are obtained similarly by subtracting parallel-polarized images from the diffuse ones. A high-resolution normal map is derived from specular images illuminated under different structured light patterns. The normal map is then embossed onto the diffuse geometry to get skin detail. Based on this approach, a photo-realistic digital character was created [Alexander et al. 2010]. Here, actor-specific blendshapes are obtained by warping a master mesh into expressive scans such that 3D geometry, surface normals and manually labeled sparse points agree. Manually sculpted geometry for the teeth and the eyes are manually added to the blendshapes for completeness. The textured model is tracked between manually-selected key poses in a multiview video setup with known lighting by applying model-based optical flow constraints. Also based on [Ma et al. 2007], Wilson et al. [2010] track dynamic high-resolution scans by combining stereo reconstruction and photometric normals. Temporally-coherent normal maps are obtained by aligning gradient illuminated images to full-on lit tracking frames both in forward and backward direction, where correspondences are computed using a novel optical flow algorithm that considers complementary gradient illumination constraints. Temporally-coherent stereo geometry of tracking frames is computed from stereo pairs using both

albedo and aligned photometric normals. Finally, the dense correspondences are used to warp and blend the texture, geometry and normals of in-between non-tracking frames.

Gosh et al. [2011] improved the method proposed in [Ma et al. 2007] and developed a low-cost fast capture system, called Light Stage X, which consists of 5 cameras. This system employs a new pair of latitude-longitude static polarizers symmetrically distributed on the sphere. This supports camera placement anywhere near the equator while providing high-quality specular and diffuse maps for computing accurate normals. High-resolution scans are reconstructed from these data by a novel adaptive domain message passing algorithm that allows for coarse-to-fine continuous optimization. Based on this work, Alexander et al. [2013] created a realtime photo-realistic digital character. Here, a detailed blendshape rig is derived from several scans and fitted to multiview video data using a novel scene flow graph. As in [Alexander et al. 2010], the eyes and the teeth are mostly sculpted manually, and the complete model is rendered in realtime with tailored BRDF models to simulate realistic skin and eye appearance.

3.1.2 Lightweight Facial Performance Capture

Lightweight acquisition methods commonly track a template or a blendshape model from either binocular stereo data or RGB-D videos containing color and depth information. Due to the low-quality and noisy input, RGB-D approaches mainly focus on the animation of believable virtual avatars using performances robustly tracked at realtime, whereas binocular approaches aim to retrieve detailed, photo-realistic meshes at high fidelity from HD stereo images.

Beeler et al. [2010] showed that their dense image-based approach (see Section 3.1.1) can also reconstruct detailed meshes at the pore level from a consumer binocular stereo camera in indoor scenarios; however, the reconstructions are limited to the visible region of the face due to the lack of priors or temporal information. In [Valgaerts et al. 2012b], the first binocular approach that tracks detailed meshes in general uncontrolled lighting setups was proposed. Similar to previous methods, a coarse template mesh reconstructed from the same stereo data is tracked using a highly-accurate image-based scene flow technique that incorporates structure-aware regularization to preserve facial features. Temporally-smoothed, yet accurate, deformations are enforced by Laplacian deformation in the temporal domain, while being consistent with the stereo images. The geometry is then refined by inverse rendering with a novel shape-from-shading framework that uses spatially-coherent local albedo information to estimate the lighting and employs temporal changes in shading cues to recover fine-scale skin details.

One of the first steps towards performance-based character animation from cheap RGB-D sensors that finds a trade-off between tracking accuracy and realtime performance is presented in [Weise et al. 2011]. Here, a novel blendshape-based tracking algorithm that combines raw depth scans and 2D texture registration with dynamic priors is used to infer motion parameters to animate avatars. To do so, two steps are performed: Model/prior acquisition and tracking. In the acquisition step, a personalized model is built by aligning a statistical shape model [Blanz and Vetter 1999] to multiple aggregated neutral scans. Then, the neutral shape is non-rigidly fitted to a set of pre-recorded user-specific scans. A personalized, dynamic blendshape model based on generic facial action units is derived from these data [Li et al. 2010], and a probabilistic animation prior modeled in the principal component analysis (PCA) space is learned. In the tracking step, the global motion of the blendshape model is estimated by ICP in a temporal window. The blendshape weights are estimated in a probabilistic framework that maximizes the alignment confidence to depth data and optical flow constraints while enforcing face dynamics consistent with the prior. Bouaziz et al. [2013]

extended this approach and proposed a dynamic expression model (DEM) that is refined in real-time. This model consists of generic blendshapes, an identity PCA model, and additional corrective deformation fields represented as the spectral basis of the graph Laplacian; the latter account for person-specific face shape and dynamics not explained by the generic model. To personalize the model, a linear expression transfer operator similar to [Sumner and Popović 2004] is used to update the blendshapes whenever the user's identity and the deformation fields change. The model's parameters are optimized in a two-step process using Gauss-Seidel: A fitting step that solves for the rigid pose and blendshape weights, and a refinement step that solves for the identity and deformation coefficients. Unlike [Weise et al. 2011], smoothed deformations are explicitly regularized by applying L_1 and L_2 norms on the coefficients. To ensure temporal consistency, the optimization is performed over the history using an aggregation scheme based on exponential decay with fixed memory overhead.

An alternative adaptive PCA model approach based on correctives is presented in [Li et al. 2013b]. Their tracking is also performed in two steps: Offline and online. In the offline step, a neutral expression of the actor is obtained by fitting a statistical model to facial scans, as in [Weise et al. 2011]. Then, generic blendshapes are transferred to the neutral shape, thereby creating the actor's blendshape model. In the online step, the blendshape model is fitted to 3D scans and detected landmarks by a flip-flop ICP-based optimization strategy that first finds correspondences and then solves for rigid pose and blendshape weights. The fitting is improved by Laplacian deformation and the meshes are further projected into the adaptive PCA model to obtain the final in-space deformation parameters. This incremental model consists of orthogonalized generic blendshapes and sufficiently far out-of-space examples, which are orthogonalized using an Expectation-Maximization (EM) based algorithm. Note that the final animation parameters are obtained by fitting the generic blendshape model to out-of-space examples.

Thies et al. [2015] presented an analysis-by-synthesis approach that reconstructs photo-realistic performances and allows for facial reenactment in realtime. This method relies on a highly-parametric face prior that models rigid pose, facial identity, and expression. It also parametrizes skin albedo and lighting to track the face in the RGB-D video streams. Contrary to state-of-the-art methods, the model's parameters are jointly estimated via inverse rendering by minimizing an energy function that considers geometric consistency to depth maps, color consistency to the face observed in the video, and shape similarity to detected 2D facial landmarks. Realtime performance is achieved by exploiting data parallel optimization on the GPU. New realistic target performances can be created by mapping expression parameters from a source performance and then re-rendering the target performance with the estimated lighting. However, their method cannot generate fine-scale skin detail (see Chapter 6 and Chapter 8). A believable mouth interior is modeled with a generic 3D teeth proxy and a 2D warping of the oral cavity.

Recently, Hsieh et al. [2015] demonstrated uninterrupted face capture in unconstrained setups showing occlusions and mild lighting changes. Their method integrates face tracking, segmentation and personalization in a unified framework. Rigid and non-rigid tracking are based on [Li et al. 2013b]. The tracked model is textured with non-occluded regions by temporal aggregation and is used as a prior to segment the face in the video. Segmentation is achieved in two steps. First, the background is discarded via distance thresholding and depth penetration tests. Second, the segmentation is refined through color-based thresholding, using superpixels for robustness. As in [Li et al. 2013b], 3D face shape is represented by an identity PCA model and generic blendshapes, but further personalized using local corrective shapes. The personalized model is gradually fitted to 2D landmarks and filtered depth data by employing a temporal aggregation strategy, as proposed in [Bouaziz et al. 2013].

3.1.3 Monocular Facial Performance Capture

Monocular capture approaches have been extensively studied in the last two decades. Here, we can mainly find methods that track sparse 2D landmarks and coarse 2D/3D parametric models. Only recently, approaches that rely on highly-parametric face priors and fully exploit dense color and shading cues have been able to acquire detailed models of a quality that competes with multiview techniques. In the following, past and present advances in the area are discussed.

Sparse Motion Capture

Methods in this category mainly detect sparse 2D fiducial points or facial landmarks that describe very discriminating facial features, or fit a coarse deformable (and textured) 2D model to images such that relevant facial features match. Most of these approaches mainly focus on sparse facial feature detection and not dense face tracking. In the following, an overview of the most relevant methods is provided. A complete survey of past and present approaches can be found in [Fasel and Luettin 2003; Zafeiriou et al. 2015].

One of the first popular approaches for sparse 2D motion tracking were the active shape models (ASMs) proposed by Cootes et al. [1995]. ASMs simultaneously generate non-rigid shape, scale and pose parameters by iteratively fitting a point distribution model (PDM) to edges of an object in the image in ways consistent with the training data. This approach requires a good initialization to converge. In [Cootes et al. 2001], a generalization of the ASM, called active appearance model (AAM), was proposed. This method fits a coarse triangulated 2D PDM using all the color information contained in the face image. The appearance and the shape (i. e., rigid pose and non-rigid deformation) of the face are modeled as separate 2D statistical PCA models from labeled and aligned training data, but they are unified into a single morphable model, as in [Blanz and Vetter 1999]. The model's parameters are obtained using an analysis-by-synthesis approach integrated into a non-linear optimization framework. Though AAMs are more accurate than ASMs, a sufficiently good initialization is needed to assure convergence. Xiao et al. [2004] showed that AAMs can indeed model 3D phenomena, albeit using larger parametrization. In this work, they recover a coarse 3D PDM from tracked 2D sequences by performing non-rigid structure from motion [Bregler et al. 2000] under a weak perspective camera model. Once the 3D PDM is estimated, 3D pose and 3D shape parameters (and their corresponding 2D AAM parameters) are jointly estimated during tracking using a fast inverse compositional approach based on a Gauss-Newton algorithm that pre-computes the Hessian matrix.

The robust tracking algorithm proposed in [Saragih et al. 2011a] improves upon ASMs and AAMs in landmark accuracy based on a 3D constrained local model (CLM) that is optimized using a regularized mean shift algorithm. This method optimizes pose and shape parameters by first estimating a response map for each facial landmark that is outputted by local SVM feature detectors, and then combining the local detectors in an optimization step that enforces a global prior over their joint motion. Asthana et al. [2013] followed a slightly different direction and proposed a discriminative boosting regression framework that learns robust low-parametric functions from local response maps to estimate the most probable shape parameter updates to fit the CLM to images. Both performance and robustness are significantly boosted by training local experts with histograms of gradients (HoG) descriptors.

Recently, Dantone et al. [2012] trained a conditional regression forest to detect a very sparse set of 2D fiducial points (i. e., mouth, eye and nose corners). This forest consists of multiple random

regression forests that learn the correlation between facial image patches and fiducial points on a subset of the data which is conditional to a particular global head pose. During testing, a set of pre-trained conditional regression trees based on the estimated probability of the global properties is selected to predict the final location of the points. Due to the reduced amount of detected points, this method cannot be used for tracking a deformable 3D model.

Xiong et al. [2013] presented an efficient non-parametric supervised descent method for sparse 2D landmark tracking in the wild. Their method learns a series of parameter updates, in the form of generic descent directions, that incrementally minimize the mean of non-linear squares functions. These functions constitute the actual fitting term, which is used to match suboptimal detected points to local SIFT detectors. Unlike AAMs and CLM based methods, their approach optimizes landmark locations directly, thus generalizing better to asymmetric gestures. Due to the uncertainty of the face contour, only features in the inner face region are robustly detected.

Contrary to previous methods that rely on an initial detection of the face [Viola and Jones 2004], Zhu et al. [2012] introduced a unified model for face localization, pose estimation, and landmark detection in the wild. The method is based on mixtures of linearly-parametrized, tree-structured part models, where each mixture accounts for pose changes. The optimal configuration (i. e., pose and non-rigid shape) is obtained by maximizing a score function that measures appearance similarity to observed features (unary potentials), spatial agreement between parts (pairwise potentials), and viewpoint consistency to a prior. The appearance evidence is obtained from local SVM detectors trained on HoG descriptors, and the non-rigid shape is represented as multiple mixture models and efficiently estimated with dynamic programming. Note that the detected facial features may represent unrealistic deformations due to the non-anatomical structure of the model.

The sparse tracking algorithms presented above normally provide facial features for model-based approaches to fit a template. Despite their robustness, they may suffer from temporal jitter and normally fall short in accuracy for high-quality face capture. As shown in Chapter 4, we build upon the state-of-the-art method proposed in [Saragih et al. 2011a] to achieve the desired level of accuracy and stability.

Also less related but very relevant are non-rigid structure from motion (NRSfM) approaches. They estimate 3D pose and non-rigid shape (often represented as a shape basis and deformation parameters) directly from 2D features tracked in an unstructured video by exploiting temporal deformation and head motion, either on the entire sequence [Bregler et al. 2000; Paladini et al. 2009] or incrementally over a sliding window [Paladini et al. 2010; Agudo et al. 2014]. In these methods, pose and shape are commonly recovered by low-rank matrix decomposition using singular value decomposition (SVD) or PCA, as formulated in [Bregler et al. 2000]. Besides, a simple orthographic camera projection is typically assumed. In [Paladini et al. 2009], a general framework for deformable and articulated objects containing missing data is presented. Pose and shape are solved reliably by an alternating least-squares factorization approach that optimally projects the solution onto the motion manifold. Paladini et al. [2010] proposed the first sequential NRSfM approach that incrementally adds new modes to the reconstructed shape basis based on the reprojection error of the learned model, where each mode is estimated by low-rank factorization. Given the current shape basis, the pose and deformation parameters at a given frame are efficiently estimated by bundle adjustment over a temporal window. In [Agudo et al. 2014], non-rigid deformations of a template mesh are linearly modeled with bending and stretching modes using modal analysis from continuum mechanics. Thus, the problem boils down to estimating the optimal pose and deformation parameters over time, as in [Paladini et al. 2010].

Model-based Motion Capture

Methods that fall in this category heavily rely on 2D or 3D coarse face models to robustly track both rigid head pose and non-rigid motion (e. g., expressions) from a 2D video. Hence, they neither replicate all the actor-specific characteristics nor reconstruct fine-scale skin details that are visible in the video sequence. These methods normally resort to image cues (e. g., optical flow constraints, a sparse set of image patches or dense color information) to track the model in the image sequence.

One of the first attempts to recover 3D (non-) rigid motion from a monocular video dates back to Li et al. [1993]. Here, an affine non-rigid motion model that encodes facial action units (AU), rigid pose and color is proposed. The model is gradually tracked to an image sequence by performing an analysis-by-synthesis feedback loop that consists of four steps: Motion prediction, model synthesis, delta motion estimation and motion correction. Having an initial estimate of the motion parameters, the first two steps deform and synthesize the parametric model based on temporal smoothness assumptions. Then, the model's parameters are corrected by computing the delta motion field (formulated as a differential approach) between the synthetic model and the observed image. Despite its robustness, the method requires accurate manual initialization to get a good estimate of the texture and shape. Similarly, Pighin et al. [1999] proposed an automatic analysis-by-synthesis approach to track face pose and expressions. This method, however, assumes an existing person-specific textured blendshape model that is fitted to images of the same actor under similar lighting conditions. Besides, their model spans a wider range of expressions by splitting the blendshapes into three subregions (eyes, forehead and lower part of the face), each independently parametrized and linearly combined by 3D morphing during synthesis. The authors showed that the estimated parameters can be directly used for generating new realistic facial animations of the actor.

In [Black and Yacoob 1995], local parametric 2D flow models for face tracking and facial recognition were explored. Assuming correctly detected eyes, mouth and brows in the first image, the method tracks these facial features using optical flow. Note that flow fields for the different facial regions are parametrized with ad-hoc models to represent global perspective motion, relative translational motion (e. g., eyelids movement w. r. t. the face) and relative curvature deformation (e. g., mouth and brows bending w. r. t. the face). These parametric models showed themselves to be useful for extracting spatial mid-level cues, which in turn allow the creation of high-level rules in the temporal domain to recognize expressions in a video.

Essa et al. [1996] described the first set of animation tools for fully automatic physically-based 3D facial modeling and tracking as well as real-time interactive animation. To accurately model realistic non-rigid facial deformations, the authors introduced a parametric physically-based model with muscle coarticulations attached to it. By fitting the models to images via automatic landmark detection and coupling them to optical flow measurements, accurate animation parameters can be obtained. They demonstrated that these parameters can be correlated to image cues from a training set to infer expressive animations from video sequences in realtime. Furthermore, they also showed that model-based flow tracking helps regularize noise and occlusions in head pose estimation. In the same line of research, DeCarlo et al. [1996] proposed an elegant model-based flow tracking method that replaces the velocities of the optical flow constraints by projected deformations of a 3D morphable model, which are parametrized by rotation, translation and non-rigid deformation parameters. The latter represent facial shape and expressions that are derived from anthropometry measurements. To determine the model's parameters, optical flow constraints are transformed into constraint forces based on Lagrangian dynamics and combined with edge-based forces (i. e., occlusion and facial feature boundary constraints) to prevent some drift propagation. Despite its efficiency, the method is limited to constant illumination and mild facial motion. Brand

et al. [2001] introduced an improved model-based flow approach based on Bayesian inference that operates directly on image gradients (not flow computations) and makes full use of image uncertainty to compute mean posterior estimates. Such improvements provide not only an accurate estimate of rigid and non-rigid model parameters but also their confidence measurements. To minimize the information loss due to inverse modeling (i. e., parameter estimation from image data), efficient reversible linear operations through chains of matrix operations are proposed. The authors showed that tracking residuals can be employed to bootstrap the shape and motion estimation to achieve subpixel-accurate tracking.

Blanz et al. [1999] created the first database of textured 3D face scans from 200 subjects and derived a PCA morphable model that represents linear variations in global face shape (i. e., identity) and albedo. They showed that a photo-realistic 3D neutral face of a person can be reconstructed by fitting the morphable model to an image in an analysis-by-synthesis flip-flop loop that iteratively compares the observed image with the rendered model. Blanz et al. [2003] extended this approach by adding a new linear basis that models simple variations in mouth deformation. Unlike Blanz et al. [1999], their method estimates illumination, focal length, head pose and model parameters in video sequences by inverse rendering using a similar flip-flop optimization loop that enforces temporal smoothness. To ensure convergence, the method requires manually labeled features and a good estimate of the head pose in the first frame. The authors showed that new animations can be created in the original video by modifying the motion curves of the model and re-rendering it under the estimated lighting. Vlasic et al. [2005] went even further and introduced the so-called multilinear face models that learn, via tensor decomposition, a set of separate, mutually-orthogonal dimensions (e. g., facial expressions, identity, and visual units of speech) from large databases of face scans. These learned models are then used to track coarse-to-medium scale, dynamic face geometry in videos exhibiting negligible head motion. The model's parameters (rigid pose and non-rigid deformation) are initialized and coupled to optical flow measurements [Essa et al. 1996; DeCarlo and Metaxas 1996] using only a subset of vertices for efficiency.

Recently, data-driven approaches for face tracking have also been explored. Cao et al. [2013] presented a novel regression-based tracking approach that learns an accurate, user-specific 3D face alignment model directly from training images. In the training step, a personalized blendshape model as well as aligned ground truth 3D shapes are constructed from user-labeled images showing a variety of poses and expressions. From these images and shapes, training samples that account for possible tracking failures are generated, and a regression function for incremental 3D facial feature alignment based on two-level boosting is learned. The regression algorithm first creates a set of appearance-based features at randomly sampled 3D points. Then, it generates and combines weak regressors, called *ferns*, that learn shape updates from image features and inaccurate 3D shape such that misalignment errors are minimized. At runtime, 3D shapes are inferred from images and previous shape estimates, which in turn are used to track the pose and expressions of the personalized blendshape model. The method proposed in [Cao et al. 2014a] improves upon this idea and learns, from a large publicity available dataset, a generic regressor that infers both 2D facial landmarks and 3D facial shape. At the heart of this method is a dynamic displaced expression (DDE) model that jointly encodes rigid pose, focal length, expressions, face shape (i. e., identity), and 2D landmark displacements. Similar to [Cao et al. 2013], the regressor learns parameter updates to incrementally fit the pose, expression and 2D displacements to new images in a sequence. In an adaptation step, identity and focal length are jointly optimized over incrementally selected frames whose estimated shapes increase the variance of the adaptive set, as proposed in [Li et al. 2013b]. Based on this method, Saito et al. [2016] recently presented a realtime, unconstrained face tracking system that explicitly segments facial regions using deep learning and processes masked RGB data for 3D

shape regression. Their segmentation approach uses a two-stream deconvolution network with bilinear interpolation and mirror convolution characteristics whose output are first concatenated and then merged into a single probability map using a final convolution layer. A final segmentation mask is obtained from the map using graph cut. The segmentation strategy is also used to generate segmented face images to train the DDE regression approach. One of the main drawbacks is the lack of temporal segmentation, resulting in high-frequency jitter during the tracking, especially when occlusions (dis)appear.

Despite the robustness and accuracy achieved by the methods presented above, the reconstructions are restricted to the parametric model, and some may suffer from drift due to fast head motion and expressive facial deformations. Besides, all of them lack fine-scale skin details, which are crucial for photo-realistic animation and editing. We address these issues in Chapter 5 and Chapter 7.

High-quality Performance Capture

The seminal work described in [Garrido et al. 2013] (see Chapter 5) presents a model-based capture approach that automatically reconstructs, in a robust coarse-to-fine fashion, highly-detailed and dynamic 3D face geometry from monocular video by leveraging detected 2D landmarks as well as flow and shading constraints. Researchers have deemed the capture of detailed 3D faces from a monocular video a very challenging and relevant problem in the digitization of human avatars, inspiring follow-up research in the field. In the following, some very recent advances are described.

Similar in spirit to [Garrido et al. 2013], Shi et al. [2014] presented a coarse-to-fine model-based approach for unconstrained capture that performs three main steps: 2D feature tracking, model fitting, and shape refinement. In the first step, candidate noise-prone 2D facial features are estimated using random forests. These features are then employed as additional fitting constraints in an adaptive AAM framework that incrementally updates the shape basis to improve landmark localization [Bouaziz et al. 2013; Li et al. 2013b]. In the second step, 3D feature positions are computed by NRSfM. A multilinear model in identity and large-scale expression variation is then fitted to the 3D positions while enforcing temporally-consistent poses and expressions between automatically selected keyframes. These keyframes exhibit the highest variation in deformation of the 3D points over the entire sequence. Finally, the face shape is refined in a shape-from-shading framework that assumes constant albedo and lighting, as described in Section 5.6. However, the albedo map is estimated at the vertex level. The refinement of the surface is performed iteratively to improve mid-scale deformations, but the improvement is minor and does not correct tangential errors.

In [Suwajanakorn et al. 2014], a simpler, yet effective, approach is proposed for total moving face reconstruction under uncontrolled imaging conditions. Their method requires neither blendshapes nor a pre-calibration step, but it assumes a person-specific average shape and an illumination-dependent appearance model reconstructed from a large photo collection available on the Internet, as described in [Kemelmacher Shlizerman and Seitz 2011]. The key component is an analysis-by-synthesis framework that combines model-based flow deformation [DeCarlo and Metaxas 1996] and shape-from-shading based refinement (Section 5.6) to align and warp the model such that it matches the observed image. To ensure accurate pose-invariant and drift-free model deformations over time, both the lighting and the pose are estimated iteratively before non-rigid registration using a RANSAC-based framework that discards occluded vertices. Temporal coherence is further enforced by considering visible correspondences of nearby frames. Fyffe et al. [2014] presented an alternative drift-free, region-based flow method that drives a template mesh in a video stream based on static textured scans. The core component is a generalized sparse performance flow graph that

connects consecutive dynamic frames in the sequence, as well as static textured scans to dynamic frames, using dense confidence constraints over different face regions. Here, sparsity is enforced by removing low-confidence static-to-dynamic flow connections. Based on this graph, the template mesh is accurately tracked between keyframes by minimizing a multi-objective cost function. This function measures the difference between projected 3D mesh deformations and optical flow vectors while restricting deformations to a spatially varying convex combination of static poses and neighboring dynamic poses in the graph; the latter are weighted by the associated flow confidences. The main drawback of this method is that it requires a static light environment map of the scene to relight the facial scans in ways consistent with the recorded video.

Cao et al. [2015] enhanced the realtime tracking algorithm presented in [Cao et al. 2014a] with local boosted wrinkle regressors that add fine-scale skin details. Local regressors are trained offline on high-quality multiview data and learn the correlation between local texture patches over the face and their corresponding wrinkle displacement maps, both parametrized in a common UV space. Patches are automatically extracted by analyzing the response probability in a precomputed wrinkle likelihood map, and they are centered and oriented along wrinkles to extract meaningful features. To enable realtime performance, local displacements in a patch are parametrized in a low-dimensional PCA space. At runtime, the mesh is accurately aligned to images via optical flow to ensure accurate and temporally-stable input features for the prediction of details. As the method relies on texture features, lighting changes, blur and occlusion deteriorate the results.

In [Wu et al. 2016], a novel anatomically-constrained local deformation model for accurate face capture is introduced. The local model is derived from personalized blendshapes and represented as a set of linear non-rigid deformation subspaces that are uniformly distributed over the face and bounded by anatomical constraints. Such constraints are given by a skull and jaw bone structure [Beeler and Bradley 2014] that limit the skin thickness w. r. t. the bones based on ground truth observations. To track the local model, the authors combine spatio-temporal and anatomical constraints to ensure globally consistent patches over time. In addition, they seamlessly blend the patches using distance-based Gaussian kernels. Even though their local approach is robust and accurate, it needs an actor-specific anatomical model to start the tracking, which is impractical for legacy videos.

A little less related is the dense NRSfM approach introduced by Garg et al. [2013a] that reconstructs dense 3D face geometry from unconstrained videos and that does not need a shape model. Here, NRSfM is formulated as a global variational energy minimization problem to estimate dense low-rank smooth 3D shapes and camera rigid motion for every frame, assuming that dense temporal 2D correspondences are given, e. g., from a flow graph [Fyffe et al. 2014] or multi-frame flow fields [Garg et al. 2013b]. Smooth, yet accurate, reconstructions are obtained by combining edge-preserving spatial regularization with a soft low-rank shape prior that ensures a compact, non-fixed shape subspace. Although their approach does not refine surface geometry, it could easily integrate shape-from-shading based refinement as a post-processing step to recover fine-scale skin details.

Despite the high-level of detail and tracking accuracy achieved by these approaches, most of them neither estimate nor parametrize personalized mid-scale deformations, such as person-specific smiles and nose shapes. Capturing such deformations not only contribute to a better tracking but also help decouple fine-scale transient details from true motion and residual misalignment, as proposed in Chapter 7. Such a separation is of particular importance when learning face rigs from tracked meshes, since artists can easily separate and control different effects in a more flexible and intuitive way, thus facilitating facial animation and video editing tasks (see Chapter 8).

3.2 Lip Tracking and Reconstruction

Accurate tracking of lips in video is deemed a problem of paramount relevance in the scientific community since accurate lip motion not only helps improve speech recognition and comprehension of the auditory channel [Kaucic and Blake 1998; LeGoff et al. 1994; Summerfield 1992] but also increases the realism of facial animations (see Chapter 9). Detecting lips in monocular videos, however, is a task hard to achieve due to the lack of depth data, disocclusions in the mouth region as well as recurrent changes in shape, color and specular reflections of the lips. Hence, lips are normally tracked in 2D using contour-based approaches that heavily rely on shape and color priors. Just recently, novel methods can reconstruct 3D lip shapes by assuming controlled multiview capture setups or large databases. An overview of these two types of approaches is provided as follows.

3.2.1 Image-based 2D Contour Tracking

Methods falling in this category mainly focus on recovering 2D contour lines for the lip boundaries (mostly outer boundaries) using either supervised learning methods or optimization-based approaches that rely on shape and/or color priors.

One of the most popular approaches for contour detection are the so-called snakes or active contour models [Kass et al. 1988]. A snake is a generic energy-minimizing spline guided by user-defined (external) constraint forces and influenced by image forces (usually image gradients) that pull the curve toward edge or line features. It is also controlled by internal forces (e. g., stretching and curvature) that resist deformation. Based on this, Bregler et al. [1994] developed an audio-visual speech recognition system that combines both acoustic cues and lip motion (captured by snakes) to improve robustness in the detection. In this framework, snakes are customized for outer lip contour detection by constraining successive deformations to lie on a manifold of plausible lip shape configurations learned from prior examples. Despite the improvement in performance, they found that image forces based on gray-scale image gradients are inadequate to detect accurate lip boundaries. Bearing this in mind, Barnard et al. [2002] adapted the standard energy function of the snake by replacing the gradient-based image forces with 2D color pattern matching templates. Here, reference templates are extracted from a neutral face at sampled points on the outer lip contour and then matched against candidates in a new frame along scan lines by using normalized cross correlation. The snake is finally pulled toward areas with correlation scores above a given threshold. In this approach, the inner lip boundary is naïvely anchored to the outer contour assuming constant lip thickness.

The methods described above only work for frontal poses and require a good initial guess to ensure convergence. To overcome these problems, Tian et al. [2000] introduced a model-based approach that fits a coarse multi-state mouth shape prior to outer lip contours by exploiting shape, color, and motion. The prior is modeled as splines and consists of three states: Open, closed and tightly closed mouth. The most probable state is inferred from the height of the lips and the color distribution inside the mouth, where the height is obtained from contour points tracked via optical flow. At a given frame, the tracked points are used to detect the mouth from which color and shape are extracted. The most probable prior is then selected and fitted to the tracked points to extract coarse lip contours. Due to the simplistic mouth model, asymmetric and expressive lip shapes are not captured. Alternatively, Eveno et al. [2004] proposed a robust and accurate lip segmentation method based on jumping snakes to avoid local minima. Assuming a rough initialization, a snake is fitted

to the upper lip contour by a succession of jumps and growing phases. The former guides the snake towards hue and luminance edge gradients, whereas the latter extends the end points. This gives keypoints for the upper lips, which are in turn used to detect the lip corners and lower lip keypoints. Then, a cubic spline is fitted to the keypoints such that it agrees with the edge gradients, thereby segmenting the lip boundaries. For each new frame, the keypoints are tracked using optical flow, and the cubic spline is updated. To prevent drift, the keypoints are refined, again with snakes guided by edge gradients, before segmentation.

Nguyen et al. [2009] tackled changes in appearance and illumination that the outer lip contour usually undergoes. They introduced a semi-adaptive active appearance model (SAAM) that incrementally adds new tracked candidates to an AAM model using a SVM detector trained to recognize aligned shapes. The incremental model consists of a fixed pre-learned PCA subspace and the new set of aligned examples, which are orthogonalized and projected to the new incremental subspace [Li et al. 2013b]. As the method discards misaligned shapes to prevent drift, it fails to detect extreme shapes.

Since the inner contour of the lips slides over and is prone to disocclusions, the methods described above mainly focus on tracking outer contours. Kaucic et al. [1998] proposed one of the first methods that tracks unadorned inner and outer contours in realtime by leveraging shape and motion priors as well as illumination-tolerant feature detectors – all learned from examples. Here, a sparse PDM model represented via B-splines serves as shape prior, whereas motion dynamics are learned from speech sequences using a maximum likelihood estimation algorithm. Contour detection is formulated as a classification problem on hue images, where a Fisher discriminant and a color mixture model identify lip-skin and inner lip contour boundaries, respectively. Based on their response maps, the inner and outer contours are deformed along the 2D curve normals in ways consistent with the priors. In this method, accuracy is sacrificed to achieve realtime performance. Similarly, Wang et al. [2004] fit an PDM model parametrized by quadratic curves to a color probability map by employing a region-based cost function that maximizes the joint probability of the partition between lip and non-lip areas. Unlike Kaucic et al. [1998], the probability map is generated from color and spatial distributions using a simple unsupervised clustering method that requires neither a feature prior nor training. As a counterpart, this simple segmentation forbids tracking of the inner contour. In both cases, the authors showed that tracked lips can improve speech recognition accuracy.

Recently, general-purpose learning-based approaches have emerged as an alternative to detecting edges and contours in images and that do not suffer from limitations of methods based on low-level features. In [Dollár et al. 2006] edge and object boundaries are identified by a supervised boosted learning algorithm, called Boosted Edge Learning (BEL), that learns to classify image pixels as (non-) boundaries from labeled examples by leveraging a large set of generic fast features over a small image patch. These features include gradients, histograms of filter responses, and Haar wavelets at different scales. As classification is performed per pixel, just a few example images provide enough data for training, but detections may be very noisy. Dollar et al. [2015] improved upon this method and proposed a generalized fast structured learning approach based on regression forests that exploits the inherent structure of edges in local patches. Here, the detection of edges is formulated as predicting local segmentation masks in input image patches. The key to efficiency is the projection of input features to randomly sampled dimensions, followed by PCA-based dimensionality reduction. The method still requires a large set of training images, which may not be available.

3.2.2 Dense 3D Lip Reconstruction

In contrast to 2D tracking approaches, recent multiview reconstruction methods attempt to extract and model the dense 3D shape of the lips in controlled studio conditions.

Bradley et al. [2010] employed a dense multiview passive performance capture system with controlled lighting that explicitly enforces accurate lip registration based on simple 2D edge detection and silhouette alignment constraints for both the inner and outer lip contours. However, the method assumes that correspondences in the inner contour region remain fixed on the mesh, thus limiting tracking accuracy. In [Bhat et al. 2013], a two-step performance capture approach is presented. This method first tracks a blendshape model using 3D mocap data and 2D contour features and then performs out-of-model refinement for improved eyelid and lip shapes by matching the contour features to dynamically selected occluding contours in the mesh. The refinement step is in turn performed in two iterative stages: First, the isoline on the mesh that has the maximum number of silhouette edges is selected as occluding contour. Then, the projected occluding contour is warped via Laplacian deformation to match feature curves on the 2D contours. Though effective, the method requires a manual selection of 2D curves in the image.

Considering previous limitations, Anderson et al. [2013a] proposed a method that uses accurately tracked 2D lip contours as additional constraints in a multi-camera photometric stereo system to improve the 3D registration of the lips. Here, a discrete snake is fitted to BEL probability maps in the image and then projected onto a reconstructed depth map to get 3D contours. Points lying on the outer contour are associated with fixed vertices in the mesh, whereas the inner contour is reckoned as an occluding boundary that is dynamically connected to predefined isolines in the mesh based on the geodesic distance to the outer contour in the depth map. In this approach, the actor needs to face the camera, and a manual initialization of the snake is required.

Another attempt to capture mouth deformations even in the presence of occlusions, fast motion, and extreme poses was presented in [Liu et al. 2015]. Here, their method efficiently fuses RGB-D video and audio data for robust, realtime face tracking. The key component is a data-driven user-independent approach based on nearest neighbor search that learns to correlate acoustic features and inaccurate lip deformations (in a parametric space) to refined mouth shapes from a large database. At runtime, refined mouth shapes are retrieved from the database based on an adaptive cost function that considers confidence and velocity changes of acoustic and user-independent lip features over a temporal window. Even though the method is robust and applicable to any subject, style and user idiosyncrasies are ignored, thus drastically limiting tracking accuracy.

Also related is the method of Kawai et al. [2015] for photo-realistic 3D inner mouth restoration of a speech animation. The method combines quasi-3D reconstruction and simulation of a generic tongue and personalized teeth with 2D appearance restoration of lip and cavity boundaries. The latter is based on a so-called *Multiview Detai-lization* algorithm that leverages a large dataset of mouth motions connected to speech to photo-realistically correct the appearance of the inner mouth (including the lips) of an animated model at a fine-grained level.

All the methods described above require multi-camera input and/or controlled recording conditions, yet many of them still struggle with strongly occluded and very expressive mouth shapes. As demonstrated in Chapter 9, we propose the first approach to capturing highly expressive lip shapes from monocular videos in general surroundings, e. g., outdoor footage or internet videos.

3.3 Face Rig and Detail Generation

Building a fully-controllable and detailed parametric facial model (i. e., face rig) is an essential element for the digitization of realistic virtual characters in feature films and computer games. Nowadays, animation artists are used to manually creating face rigs of actors with custom-designed control parameters. Facial expression control is normally achieved by using a set of blendshapes that span intuitive atomic face expressions and that are linearly combined to obtain a new pose [Lewis et al. 2014]. Alternatively, physics-based muscle models can be used for animation control [Sifakis et al. 2006], either separately or in conjunction with a blendshape model (see Section 2.1.2 for more details). The automatic creation of fully-controllable parametric models that capture the mannerisms, expressions and details of the human face has been a major concern in the face capture and facial animation community in an attempt to try to streamline the effortful manual process done by artists.

One of the first endeavors to model the global facial anthropometry across people from a database of 200 laser scans was proposed in [Blanz and Vetter 1999; Blanz et al. 2003]. The authors modeled the global variation in shape and appearance across the samples using PCA. Such an identity PCA model constitutes one of the base coarse components in the multilayer face model presented in Chapter 7, as well as in the automatic face rig generation introduced in Chapter 8. Multilinear models presented in [Cao et al. 2014b; Vlasic et al. 2005] go one step further and try to capture the facial anthropometry on different levels of variation (e. g., expression, identity, and mouth shape) by learning via tensor decomposition a set of mutually orthogonal dimensions from a large database of expressive facial scans of different individuals. Fitting a personalized parametric expression and identity model to the face of an actor with the representations mentioned above, however, is a challenging problem and also counter-intuitive, since they have control dimensions that are often of global support and lack semantic interpretation. Therefore, such models cannot be readily used by artists for rigging. To overcome these problems, Tena et al. [2011] presented a linear piecewise modeling method that learns a collection of PCA local models from facial scans. These local models are independently trained on different facial regions but share common boundaries to enforce global consistency. They showed that the region-based formulation not only generalizes better than the holistic PCA counterpart when fitted to unseen data but also gives the user a localized manipulation control via *click-and-drag* interaction to create new animations. However, such localized models provide a segmentation into fixed parts that may lack semantic control, as opposed to the flexible localized control built into blendshape models designed by artists.

Generic blendshape models personalized to the actor's face have been extensively used by different facial performance capture methods, both in monocular camera settings [Cao et al. 2013; Cao et al. 2014a; Saito et al. 2016; Thies et al. 2016] and RGB-D sensor setups [Weise et al. 2011; Thies et al. 2015]. Face personalization is achieved by deforming the model into a set of facial scans or by directly fitting it to RGB(-D) video, either during tracking or in a pre-processing step. Nevertheless, such generic blendshape adaptation fails to capture person-specific expression details performed by the user. As such, some recent approaches estimate not only identity and blendshape parameters from the captured facial performances but also person-specific correctives on top of the coarse model to give a better personalization [Bouaziz et al. 2013; Li et al. 2013b; Hsieh et al. 2015]. The main drawback of these approaches is the need for both depth and optical input data. Latest advances in monocular facial performance capture now allow for capturing detailed, dynamic 3D face geometry (see Section 3.1.3), but they are unable to intuitively parametrize person-specific idiosyncrasies in expression, and to learn a model for wrinkle generation that nicely correlates to blendshapes or coarse expressions. Detailed blendshapes can be alternatively captured from complex 3D

facial scanning systems and used later to approximate details for newly synthesized expressions [Alexander et al. 2013; Fyffe et al. 2014]. However, such a strategy may fail to reproduce all the nuances and face dynamics of an actor, requiring more exemplars or extensive manual interaction.

One of the first attempts to learn models for wrinkle synthesis goes back to Bickel et al. [2008]. Here, coarse-scale facial performances captured from mocap systems are augmented in realtime with a fine-scale detail layer, which is learned by correlating sparse measurements of coarse-scale skin strain with a fine-scale detail layer (skin bulges and wrinkles) from a small set of example poses. In this approach, details are parametrized as local displacements, whereas skin strain is computed as the relative stretch of edges in a sparse feature graph, where its nodes correspond to point locations in the coarse mesh. Thus, the learning problem is formulated as a scattered data interpolation in pose space for which radial basis functions with biharmonic kernels are employed as interpolators. The authors showed that new realistic animations can be interactively created using mocap data or sparse handlers on the mesh. Similarly, Ma et al. [2008] proposed an approach that learns the correlation between coarse-scale skin strain from mocap data and high-resolution detail layers captured from a 3D scanning system. However, details are represented as two independent layers: Medium-scale face dynamics and fine-scale deformations at the pore level. The former is represented as a 3D displacement map, while the latter is encoded as a height map that embeds displacements along the normal direction. Consequently, coarse-scale skin strain is projected into the UV-space and parametrized by biquadratic polynomial displacement maps. Two linear maps are then learned from the data, which are then used to synthesize medium- and fine-scale details on coarse meshes tracked by mocap data. In [Bermano et al. 2014], low-resolution art-directed facial performances are also enhanced by generating actor-specific expressiveness and details learned from a large corpus of high-resolution example meshes that are captured in a multiview camera system. These examples represent short, yet expressive, subsequences. At the heart of the approach is a convenient shape space representation that parametrizes coarse-scale deformation and detailed expressiveness of samples in the database via deformation gradients. During synthesis, new coarse meshes are simply projected into the shape space to find the optimal linear combination of examples that match the new expression. Once found, the inferred weights are then used to generate a detailed expressive layer for the coarse mesh.

The main drawback of the wrinkle generation methods mentioned above is that the fine-scale detail layer is mainly driven by coarse 3D geometric deformations that are not always intuitive to control and animate. Besides, such approaches can mainly be applied to performance capture data obtained from controlled and complex camera systems.

Recently, some approaches can synthesize details on arbitrary facial performances captured from commodity sensors, e. g., Kinect or webcams. Li et al. [2015b] proposed a lightweight wrinkle synthesis method for enhancing coarse facial models captured from an RGB-D camera. The method is divided into two stages: Wrinkle extraction and generation. In the first step, local wrinkle exemplars in the form of gradient images and height maps are extracted from pre-computed high-quality textured meshes. In the second step, given a gradient map computed from an input face image, the optimal synthetic height map is obtained in an EM framework. First, it estimates synthetic gradient and height maps given fixed exemplars. Then, it finds the optimal exemplars from the database using the input gradient values and the estimated height map. The resulting height map is then used to compute a wrinkle layer for the coarse 3D face model reconstructed from the RGB-D sensor. As a by-product, they showed that a detailed blendshape model can be created from static expressions performed by the actors. Similarly, Cao et al. [2015] enhanced monocular reconstructions with local boosted regressors that add fine-scale skin details by leveraging a database of high-quality multiview data. Contrary to Li et al. [2015b], the local regressors learn in an offline step the actual

correlation between local texture patches over the face and their corresponding wrinkle displacement maps. This way, wrinkles can be quickly regressed at runtime from the learned model. As both methods rely on photometric cues, lighting changes, blur and occlusions deteriorate the quality of the results. Besides, they cannot directly generate wrinkles from blendshape weights, which is a de facto standard in facial animation.

So far, the automatic creation of fully personalized and detailed face rigs remains unsolved. Li et al. [2010] introduced an automatic method for generating actor-specific blendshape rigs from example expressions. These rigs preserve the semantics of a prior model, yet they capture the mannerisms of the target actor. The optimal set of target blendshapes is formulated as an optimization problem in the gradient domain [Sumner and Popović 2004] by interleaving between two steps. The first step solves for target blendshapes whose linear combination optimally match the target examples in a least squares sense while preserving the semantics of the prior. The second step keeps the target blendshapes fixed and optimize for the linear blendshape weights. The main drawback is that detailed example poses must exist beforehand and correspond to valid blendshape combinations of the prior; otherwise, person-specific characteristics are not transferred.

Ichim et al. [2015] proposed an approach that reconstructs a person-specific face rig with personalized albedo map and dynamic wrinkles directly from hand-held monocular cameras in controlled setups. Both a static and dynamic modeling step is performed to capture the rig. In the first step, a generic blendshape model is fitted to a structure-from-motion-based reconstruction of the head at rest, and an albedo map is obtained from multiple views via intrinsic decomposition and Poisson integration. In the second step, the generic blendshape model is tracked over a sequence of actor-specific expressions, which exercise the different dimensions of the blendshape model. Then, the model is refined by allowing out-of-space deformations in expression and depth, using both flow and shading constraints. Transient fine-scale details are additionally recovered by applying shading-based refinement and stored as normal and occlusion maps. Finally, a generative bump map is learned, which correlates strain of facial features in the model with displacement maps. The result is a detailed face rig that can be animated with blendshape parameters. Note that several steps require manual intervention and the approach is unsuitable for legacy video.

Chapter 8 presents the first automatic approach that creates a fully-personalized parametric face rig, which is composed of a generic identity and blendshape model at the coarse level, a corrective personalized layer at the medium level, and a fine-scale generative detail layer. Unlike previous methods, our approach reconstructs detailed face rigs from uncontrolled monocular setups, i. e., it requires no initial 3D scan nor a set of prescribed facial expressions. Moreover, the reconstructed face rigs can be intuitively controlled by an artist by simply manipulating blendshape controllers, which trigger all levels of details automatically.

3.4 Speech-driven and Video-driven Facial Animation

Facial animation has been studied for decades in the graphics community and aims to animate 2D/3D computer generated models of characters with plausible mouth coarticulations and facial motion. The characters can have human-like shape or be a fantasy creature. Such characters can be used as avatars in video games and movies (see Section 3.1), or even in VR scenarios and teleconferencing. The literature is quite extensive, and in this section, we only focus on methods that drive facial models through speech or video to produce plausible visual animations of a random virtual character, i. e., not a detailed replica of the user that accurately mimic his/her facial expressions and

mannerisms, as described in Section 3.1.

3.4.1 Speech-driven Animation

Methods in this category typically associate units of sound (*phonemes*) to their visual counterpart (referred to as *visemes*) to drive a parametric animatable model, where visemes are usually represented as motion curves of fixed or variable length in the parameter space. The mapping between audio and visual cues can be learned from a corpus of mocap data connected to speech or by correlating existing model parameters with speech.

Brand et al. [1999] proposed one of the first methods for voice puppetry that steers the whole range of face dynamics of an avatar directly from speech by learning a mapping from phonemes and prosodic features to facial motion trajectories performed by multiple subjects. To do so, the audio and mocap data from a video are first modeled as a probabilistic state machine based on a hidden Markov model (HMM) that maximizes information gain. With the entropic estimation, a dynamic facial model is learned from the mocap data, which in turn is used to estimate the probability of audio features to each facial state, thus resulting in a vocal HMM that contains face dynamics. During speech synthesis, the most likely facial states are predicted and then mapped to motion curves to drive the avatar.

In [Sifakis et al. 2006], a physically-based approach is proposed, which simulates 3D speech animation of a person-specific facial muscle model using learned audio-controlled activation curves. During training, phonemes are directly associated to muscle activation curves (*physemes*) inferred from mocap data. Given a new speech signal (or a sequence of phonemes), the optimal sequence of physemes is found by solving a global optimization problem that maximizes phoneme match while accounting for discrepancies between selected physemes in overlapping regions. The authors showed that facial expressions and external forces can be easily integrated into the physics-based model to increase realism.

Blanz et al. [2003] proposed a very simple approach that directly maps detected phonemes from text or speech onto static visemes in a database assuming 1-1 correspondences between audio and visual units. Natural in-between viseme deformations are then enforced in a post-processing step by applying keyframe interpolation with cosined-shape-acceleration functions, followed by temporal blur. Kshirsagar et al. [2003] synthesize natural animations with coarticulation effects by matching phoneme streams to *visyllables* (visual counterpart of syllables), represented as visemes of fixed length. In the training step, syllables are automatically extracted from phoneme streams (syllabification) and transformed into clusters according to individual articulation rules for consonants. Visyllables are finally associated to clusters and encoded as fixed-length motion curves. To ensure continuity between visyllables in the database, the motion curves are manually altered to agree at the boundaries, either by skewing or smoothing operations. As a result, new animations can be created from speech by direct visyllable lookup in the database. Ma et al. [2006] improved upon the previous matching and synthesis strategy by searching and concatenating optimal variable-length utterances in a large corpus of mocap data. In the search step, utterances are selected from a large motion graph based on the following criteria: They agree at boundaries, are as long as possible and match the input speech. To ensure natural and smooth concatenations, a trajectory-smoothing algorithm is further applied. In [Taylor et al. 2012], an alternative approach is presented, which learns dynamic, concatenative visemes to render coarticulated speech animation. Contrary to previous methods, the speech corpus is segmented and clustered into consecutive visemes of varying length using only visual gestures, represented in this case as mouth motion parameters of an AAM.

Similar to Ma et al. [2006], dynamic visemes in a motion graph are optimally concatenated by minimizing discontinuity and phoneme mismatch constraints. Timing is also considered to enforce synchronized lip motion.

Cao et al. [2004] addressed performance issues of data-driven methods that are based on large graph motions with exponential complexity (see above). To do so, they proposed an efficient data structure, called *Anime Graph*, that keeps track of the recording order of clustered visemes with a directed graph and that further associates phoneme labels to nodes in the graph via indexing. To efficiently search nodes in the graph, the input phoneme stream is split into temporal chunks and greedily matched against viseme subsequences using only two criteria: Phoneme matching and number of jumps in the graph. Linear blending and smoothing are further applied to produce jitter-free speech animations. Berger et al. [2011] further contributed with a generic modular framework, called Carnival, that combines text- and speech-based processing tools with realtime speech animation. This framework, for instance, allows users to define tailor-made animation controllers to drive an avatar from a speech signal, and to customize signal processing tools to extract more sophisticated acoustic parameters or speech categories.

The main problem of the methods described above is that they try to predict the entire range of facial motion solely from a speech signal, which is not always possible [Yehia et al. 1998]. Bearing this in mind, in [Deng and Neumann 2006] a novel data-driven animation system for expressive animation synthesis and intuitive editing was proposed. At the heart of the method is an audio-visual mocap database recorded in four different emotions and represented as a multidimensional motion graph. Given a new phoneme-aligned speech, a constrained dynamic programming algorithm creates new smooth speech sequences by minimizing a cost function that jointly considers dissimilarity to input phoneme streams, velocity change of selected phonemes at boundaries, and user-defined constraints, such as emotion specifiers and motion-node constraints. Note that user constraints are specified via an intuitive and flexible phoneme-isomap interface that can add and delete nodes in the graph. Similarly, Anderson et al. [2013b] proposed a visual text-to-speech system that also allows the inclusion of emotion content to generate expressive animations. However, their approach leverages a larger audio-visual corpus containing six different emotions, where visemes and emotions are parametrized by an extended AAM that can separate rigid pose from localized facial deformations. Here, a cluster adaptive training (CAT) based on an HMM is employed to correlate fixed-length phonemes to visemes, while allowing the user to modulate emotion via expression weights. The key component of CAT is the use of an ensemble of decisions trees, each capturing speaker-dependent information at a different emotion level. Hence, expression weights and speech can be combined at runtime to drive an expressive model with coarticulated speech.

In Chapter 6, we also show that a strong coupling of high-quality performance capture data and speech analysis also leads to plausible expressive animations with coarticulation effects.

3.4.2 Video-driven Facial Animation

Here, we can find methods that, given a video of a user (source), extract animation parameters or transfer tracked facial motion from the source performance to animate a 2D/3D avatar (target). In the literature, the animation of avatars is usually referred to as facial puppetry or cloning.

Parameter-based Transfer

In this category, we find methods that directly transfer parameters between a source and a target model with semantically similar expression bases, and approaches that learn in a training step a linear mapping between the source and the target parameters to enable expression transfer at runtime.

Chuang et al. [2002] proposed an offline 3D facial animation approach that combines automatically detected 2D mocap data and blendshape interpolation. This method is described by two main steps: Decomposition and retargeting. In the first step, the 2D mocap data is decomposed into a weighted combination of keyshapes, which are automatically selected by analyzing the maximum spread (variance) of 2D shapes in the PCA space. After keyshape selection and weight extraction, 3D blendshapes of an avatar that resemble the keyshapes are manually created. In the retargeting step, the weight curves are simply transferred to create an interpolated facial animation for the avatar. Similarly, recent performance capture approaches track blendshape models of a user from monocular footage [Blanz et al. 2003; Cao et al. 2013; Cao et al. 2014a] or RGB-D sensors [Bouaziz et al. 2013; Li et al. 2013b; Weise et al. 2011] and then transfer the estimated blendshape coefficients directly to artistically-created non-human models that hopefully share the same semantic expressions of the user's model. Hence, the quality of the facial animation greatly depends on the modeling skills of the artist.

One of the problems of the methods described above is that the cloned expression appears as the user's expressions imposed onto the avatar [Theobald et al. 2009]. Even worse, due to possible semantic discrepancies between the identity of the user and the avatar, the user's expression may lie in a different place in the avatar's space (even when the expressions share the same semantic meaning about the mean). While this works for cartoon-like avatars, it may produce non-realistic expressions or artifacts on human characters. In Chapter 6, we demonstrate this is the case and show a simple, yet effective, approach that aligns the origin of the source and target parameter space to transfer plausible facial deformations.

Multilinear models presented in [Cao et al. 2014b; Vlasic et al. 2005] also help overcome this problem by learning from a large database of faces a set of mutually orthogonal dimensions (e. g., identity and expression variation) via tensor decomposition. This way, expressions can be completely decoupled from identity. As a result, tracked expressions coefficients (obtained by aligning the multilinear model to optical flow constraints, landmarks, or any other visual cue in a video) can be directly transferred across reconstructed models of individuals without bias. Alternatively, [Weise et al. 2009] introduced a live facial puppetry approach that learns from a set of source and target training shapes a linear subspace that optimally spans the source expressions in the target space. Having this linear map, new source expression weights (estimated by fitting the source model to facial scans [Weise et al. 2007]) can be directly used to generate target expressions in real-time. Theobald et al. [2009] proposed a simpler approach for mapping expression parameters across reconstructed AAM models without the need for a sophisticated capture or a large database. They consider expression cloning as a geometric problem. Assuming that expressions are parametrized by a linear shape basis, they precompute the inner product between a basis vector in the source model and the corresponding basis vector in the target model and use it to weigh the expression parameters transferred from the source to the target model.

Saragih et al. [2011b] presented a realtime facial puppetry system that requires only a single neutral image of a user and an avatar. The system consists of two phases: Offline and online. In the offline phase, a model for the user and the avatar are created by first fitting a parametric 3D shape model to 2D landmarks detected on the input images and then by transferring a discrete set of average

expressions from a generic basis of shape and texture variation. With the generated expressions and the shape basis as regularization, an optimal mapping between the user and the avatar model is learned. Texture variation is also learned from the generic basis and correlated to expression changes. At runtime, the user's face is tracked, and the estimated parameters are transferred to the avatar using the learned mapping.

The main drawback of learning based methods is that they need a large set of source and target training examples with similar expressions to infer a meaningful mapping function. However, such data may not be readily available. In Chapter 8, we deal with this problem by learning from high-quality monocular reconstructions a personalized face rig that correlates standard blendshape parameters to person-specific expressions and face detail. This way, target face rigs can be animated with parameters estimated on a source performance, while preserving the idiosyncrasies of the target character.

Motion-based Transfer

Approaches that fall in this category map either source 2D/3D facial motions captured from video or vertex deformations of a tracked source model onto a target model, provided that dense correspondences of some sort are available.

Eisert et al. [1998] proposed one of the first approaches for visual puppetry in teleconferencing scenarios. In a preprocessing step, a coarse parametric textured model of a user (represented via splines) is derived from 3D scans. At runtime, the textured model is tracked in a monocular video using model-based optical flow constraints, as in [DeCarlo and Metaxas 1996]; however, dense correspondences are computed between the tracked textured model and the current image in a hierarchical loop framework to boost performance. The digital model of the user can be then rendered in arbitrary virtual scenes under user-defined camera positions.

Noh et al. [2001] introduced a flexible expression cloning approach that can reuse dense 3D motion vectors of a source model (acquired from mocap data or a video) to create similar animations on a target model. To establish dense correspondences between the models, their method requires a few sparse correspondences and performs volume morphing with radial basis functions, followed by a cylindrical projection for a full surface match. To account for geometric differences in the transfer, per-vertex deformations are parametrized in the local coordinate system of each vertex, and their magnitude is locally adjusted relative to the bounding box size of neighboring vertices. Here, heuristic rules are applied to select sparse constraints and deal with correspondences in the lip area. Also quite related is the popular general-purpose deformation transfer approach for triangle meshes introduced in [Sumner and Popović 2004]. This method parametrizes local triangle deformations of a source and a target mesh as affine transformations and maps deformations induced by the source mesh onto the target reference mesh by solving a global optimization problem that assembles locally disconnected vertices in the deformation gradient domain, akin to Laplacian-based surface editing [Sorkine et al. 2004]. Unlike Noh et al. [2001], dense correspondence computation is integrated into the deformation framework and performed via an iterative ICP-based registration algorithm, which is guided by a set of predefined sparse correspondences.

Based on [Noh and Neumann 2001; Sumner and Popović 2004], Song et al. [2007] proposed an expression transfer framework for meshes and images; in the latter case, it can transfer facial detail (e. g., folds), even in the presence of lighting differences. The key component is a vertex-tent coordinate (VTC) representation that encodes deformations of a vertex relative to its one-ring neighborhood as well as the surface normal. As in [Sumner and Popović 2004], expressions are then

transferred via integration in the gradient domain. In the case of images, shape transfer is carried out via image warping using detected landmarks. To account for high-frequency texture changes, the images are aligned and represented as 3D image grids, where the z coordinate represents the pixel luminance. VTC-based expression mapping is then performed to compute the pixels' luminance values, which in turn correspond to changes in shading. Due to the simple image registration approach, temporal flicker is expected in video-based transfer setups.

Chai et al. [2003] showed that rich 3D facial animations can also be created live from 2D facial features and a database of 3D mocap data. Their method is divided into two steps. In an offline step, the user's mocap data is parametrized in a low-dimensional space by separating pose from non-rigid shape deformation (shape basis + weights) via NRSfM. Here, a 1-1 mapping between shape weights and coarse motion control parameters from tracked 2D facial features is also established. These motion controllers represent robust translation invariant features, e. g., relative mouth deformation and eyelids movement. In an online step, shape weights are estimated from motion control parameters by nearest neighbor search over a sliding window for robustness. Finally, the user's shape is computed via weight interpolation in the gradient domain and mapped onto the avatar using precomputed dense correspondences, as described in [Noh and Neumann 2001].

Recently, photo-realistic 3D avatars that retain the face shape and details of a target actor have also been animated from source videos [Suwajanakorn et al. 2015]. The key element is a large photo collection of a source and a target actor from which textured models are derived. Dense correspondences between the target and the source textured models are established via optical flow in a normalized appearance and illumination space, and source coarse expressions are transferred to the target model using a magnitude-adjusted motion field. To reproduce appearance consistent with a new source expression in a video, their method computes a weighted average of images in the target photo collection, where the weights represent the color similarity as well as the confidence of high-frequency details at different image resolution levels. Inconsistent illumination in the target photo collection is handled by preferring uniform color distributions in the lower resolution levels.

Even though some methods described above can transfer detailed motion fields to a target model in a flexible way, they do not provide an intuitive parametrization to control the avatar's facial motions, e. g., through blendshape controllers. As such, these methods are often not used by animation artists for interactive editing tasks. In Chapter 8, we contribute such an approach that can animate detailed personalized avatars with intuitive generic controllers that steer facial expressions, while still reproducing person-specific idiosyncrasies.

3.5 Face Replacement and Rewriting in Video

Editing faces in images, and particularly in videos, may be reckoned as one of the most important steps in the digitization pipeline since the final composite must simply look realistic to get the viewer's acceptance. As such, this task is carried out by trained artists and requires lots of effort and time. In the literature, several methods have been developed to try to automatize this laborious process without sacrificing too much visual quality. Here, we can recognize two main categories for face editing: Replacement and rewriting. Replacing faces can be useful for online identity protection and has been applied in movies for actor replacement (this also includes the same actor with substantially different facial features, e. g., a younger or older self). Rewriting the face content is interesting for dubbing, retargeting and video montage scenarios.

3.5.1 Face Replacement

Most of the methods in this category try to exchange faces (possibly with different illumination) in images or videos, such that the synthesized result looks sufficiently realistic. Note that the target face may (or may not) show the same expressions as the source.

One of the first attempts to replace faces in images with different lighting and head pose dates back to Blanz et al. [2004], where a statistical 3D face prior [Blanz and Vetter 1999] is fitted to a source and target face to extract the head pose, face shape, skin color and illumination parameters. With the extracted information, face replacement is then achieved by rendering the reconstructed 3D source model in the target image under the 3D pose, color gain, contrast, and illumination estimated on the target face. To generate the final composite, seams across the face boundaries are removed by alpha blending, and scalp hair is manually segmented and synthesized on top of the rendered face. The main drawback is that the method requires manual initialization to ensure a good estimation of fitting parameters, and thus a realistic composite.

Bitouk et al. [2008] proposed a fully-automatic image-based system for replacing an input face image with potential candidates in a large photo collection downloaded from the Internet. To efficiently select candidates, the database is clustered into bins of similar pose in a preprocessing step. At runtime, candidates that match the input pose are chosen based on the similarity in blur properties as well as albedo and lighting. The latter are obtained via intrinsic image decomposition assuming a Lambertian model. Color differences in the replacement boundary are considered as an additional selection criterion to avoid seams in the final composite. Finally, the candidates are warped into the input image using detected landmarks, and lighting differences are adjusted using the ratio image formulation proposed in [Liu et al. 2001]. Although the method achieves high-quality results, it is limited to the accuracy of the landmark detection and pose variability in the database.

The methods described above can produce very realistic synthetic faces in single images. Their application in videos, however, is not clear and may require additional temporal constraints to regularize undesired pose and expression variation and enforce synthetic faces with similar identity over time.

The traditional way to replace faces in movies is to first acquire a dense, high-quality 3D textured model of an actor in complex and expensive professional setups (see Section 3.1.1). The 3D model is then reanimated by artists or through performance capture data and rendered back into a target video under target scene lighting. Finally, the rendered geometry is blended in with the actor's face in the target video. Borshukov et al. [2003] employed the Universal Capture system to acquire a 3D model of an actor. The model is manually animated and tracked in the scene. Here, realistic subsurface scattering effects are simulated via blurred 2D lightmaps to render a photo-realistic skin texture. In [Alexander et al. 2010], a 3D model of an actor is obtained using the Light Stage 5 capture system, which reconstructs highly-detailed diffuse, specular and normal maps for photo-realistic rendering [Ma et al. 2007]. The model is then tracked semi-automatically in a video and rendered under known lighting conditions. In both approaches, the digital model depicts the same actor and is relit with the incident illumination captured from a light probe. Jones et al. [2008] proposed a method that is capable of replacing two different faces in a video. To do so, both the source and target face models are acquired with the system proposed in [Ma et al. 2007] and tracked in a marker-based capture setup with uniform lighting. To allow high-quality relighting of the source model, specular and diffuse albedo components are estimated using a dichromatic color model-based separation technique in the SUV color space. Finally, the tracked source model is rendered into the target video with the estimated target pose. Note that the target face must stay at

a neutral pose during the whole performance.

Model-based approaches for high-fidelity video replacement are hard to deploy and typically require significant manual intervention. In view of these problems, Dale et al. [2011] presented an approach that works on monocular video, requires minimal user interaction, and accounts for differences in shape and expression between a source and target face. Here, both performances are tracked with a multilinear model as in [Vlasic et al. 2005], and the corresponding geometry is employed to align the source with the target performance, both in the spatial and temporal domain. Temporal alignment is performed via dynamic warping based on the mouth motion, while spatial alignment is done by rendering the source model in the target pose space. Temporal consistency of face boundaries in the final composite is maintained by computing the optimal seam in a graph cut framework, such that pixel differences between the source and target faces are globally minimized while enforcing consistent pixels between consecutive frames. Although the results are impressive, user interaction is still needed, and performances must share similar illumination, timing, and pose.

In Chapter 4, we propose a fully automatic image-based approach for video face replacement that excels in simplicity, is robust to changes in head pose, and produces convincing results in arbitrary videos. Unlike the methods described above, our approach does not need complex 3D models, or a large database of faces, or source and target videos with similar scene content.

3.5.2 Face Rewriting

Methods in this category alter the original facial content (expressions and mouth motion) performed by a target actor in a video, using either speech or facial motion transferred from a source actor. Unlike face replacement methods, the target actor preserves his/her identity.

Speech-based Rewriting

Similar to speech-driven animation techniques, these methods associate phonemes to visemes to generate new video animations; however, visemes are explicitly represented as short video segments or images. The audio-visual map is learned from an actor-specific corpus and applied to the same actor, thus preserving his/her facial dynamics and mannerisms. The main challenge is to produce new believable videos with correct motion, dynamics, and coarticulation effects.

One of the first fully-automatic video rewriting approaches can be found in [Bregler et al. 1997]. Here, a video corpus is reordered and aligned to match an input speech track while being consistent to the original performance in regions outside the mouth. In a preprocessing step, the video corpus is tracked and broken down into triphones to capture coarticulation effects. Visemes are directly associated to triphones and encoded as robust features that describe the lip shape. In the synthesis step, the optimal rearranged triphone videos are found by minimizing a cost function that penalizes dissimilarity of phoneme context and mismatch between selected lip shapes in overlapping segments. The triphone videos are then shifted and skewed to ensure shape continuity and proper timing. To generate the final composite, the upper face in the triphone videos is globally aligned to the original sequence via 2D affine warping, and the boundaries of the inserted mouth images are cross-faded with the original mouth shape. Note that the method only works for the same actor and language. Besides, it only succeeds for simple head poses.

Ezzat et al. [2002] formulated video-realistic speech animation as a learning problem. Based on a set of prototypes selected from a large corpus, their method learns a multidimensional morphable

model (MMM) that parametrizes the space of all possible texture and motion flow configurations w. r. t. a neutral face to synthesize unseen mouth motions. New configurations are created by interpolating motion flow prototypes in a sparse flow graph and warping prototype images with the interpolated flows. After parametrizing the corpus in the MMM space, the linear correlation between phonemes and parameter trajectories as well as the level of smoothness between phonemes is estimated using gradient descent learning. Then, this map is used to predict trajectories from new phonemes. The final composite is created as in [Bregler et al. 1997], but face alignment is performed via optical flow. Chang et al. [2005] extended the previous approach to create new speech animations when only a small corpus is available. To this end, an MMM previously constructed from a large corpus is transferred to the limited corpus, and their associated MMM phoneme models are adapted to preserve the speaking style of a new person. The transfer step is achieved by simply finding, in a least squares sense, new optimal prototypes in the small corpus whose flow field and texture match those of the pre-learned MMM. In the phoneme adaptation step, new representative images in the smaller corpus are selected to learn an optimal intermediate linear map that projects the already learned distribution of trajectories in the new MMM space.

Liu et al. [2011] proposed a system that can synthesize new expressive mouth animations in an existing background sequence. To this end, a video corpus with neutral and smiling mouth expressions is recorded and aligned. This corpus is then associated with phoneme and expression tags, appearance parameters of an image in the PCA space, and robust geometric parameters (mouth width and height). The optimal concatenation of mouth images is found by minimizing a weighted cost function that measures the distance to a given input phoneme context, the difference between selected mouth images, and the dissimilarity to an input expression tag. The selected mouth images are finally aligned and synthesized on top of a background sequence whose face is in a neutral pose.

Although the methods described above can generate mouth animations with good motion dynamics and proper coarticulation effects, they produce dull and inexpressive animations, cannot handle fast or complicated head motion, and are restricted to the language spoken in the corpus. Chapter 6 addresses all these issues.

Motion-based Rewriting

Most motion-based rewriting methods transfer facial motion from a source to a target face by utilizing either normalized motion fields or motion parameters, or even by simulating motion via image-based interpolation of candidate frames that are selected manually or based on their similarity to source faces.

One of the first methods for transferring expressive facial motion from a source to a target face image was presented in [Liu et al. 2001]. This method enhances traditional motion-based geometric warping by additionally capturing and transferring changes in shading that arise from subtle skin deformations. Assuming Lambertian reflectance and aligned faces, shading effects that appear in a source expression are mapped onto a neutral target face using the so-called expression ratio image (ERI), represented as the quotient of an expressive and neutral face. As the quality of the results highly depends on the alignment strategy, temporal stability is not guaranteed and local illumination changes on the face, which is common in a video, may violate the quotient formulation.

In [Kemelmacher-Shlizerman et al. 2010], the facial motion of a source video character is indirectly transferred to a different character in a target video by rearranging and roughly aligning target frames, such that the selected faces match the expression of those in the source video, akin to the face replacement method of Dale et al. [2011]. However, in this case, face pose in both sequences

may differ, and so may the lighting and expressions. To handle this, the authors proposed a simple greedy approach that selects frames by measuring the dissimilarity in appearance and pose over a small temporal window, where the face appearance is represented as local binary patterns (LBP) descriptors computed on the mouth and eyes region. The rearranged sequences still suffer from temporal jitter, which can partially be controlled using long target videos with diverse expressions.

Also related is the approach proposed by Kemelmacher et al. [2011] that automatically creates a smooth transition of aligned photographs between a source and target input face. The main component is a large photo collection of a person with varying expressions at different points in time. Here, the photo collection is represented as a face graph of aligned images, where edges in it describe the similarity in appearance and pose between frames as well as temporal proximity between images (in case timestamps are available). Unlike the previous method, the appearance is parametrized as HoG descriptors. A smooth transition of images can be then found by traveling the shortest path from the source to the target face and then cross-dissolving between selected images. However, this causes noticeable ghosting artifacts in facial regions.

Inspired by the limitations of previous approaches, Li et al. [2012] proposed an image-based system that generates realistic facial animations for a target actor by optimally retrieving candidate faces from a target database with similar expressions to those in a source sequence. Unlike previous retrieval approaches, both expression similarity and its velocity change are used as criteria to choose target candidate frames. Here, expressions and velocities are represented as normalized flow fields that measure the optical distance to the neutral frame or between consecutive frames, respectively. A directed face graph with candidate frames is then constructed, and the optimal retrieved sequence is found by traversing it, while enforcing selections with minimal frame jumps. As the database may potentially lack expressions, retrieved images are blended in with ERI reconstructions [Liu et al. 2001] to add missing motions and shading details. Despite the high-quality animations, the method only handles frontal poses and slow motions.

One of the main limitations of the methods described above is that they require large databases or long sequences to produce believable facial animations. Furthermore, source and target poses must usually be very similar to ensure artifact-free facial transfer. Still, jitter or ghosting artifacts cannot entirely be prevented. In Chapter 4, we introduce a robust image-based face transfer approach that requires no dedicated database, but just a short sequence with arbitrary expressions and moderate head motion.

Capturing and transferring detailed facial dynamics as well as rendering photo-realistic synthetic sequences may still be difficult for image-based methods, especially in the presence of challenging head poses and facial deformations. In Chapter 6, we extend the image-based approach presented in Chapter 4 and leverage high-quality model-based face tracking and scene lighting estimation to transfer plausible facial dynamics and render new photo-realistic synthetic sequences, both crucial for dubbing scenarios in movies. Inspired by this work, robust model-based techniques for real-time photo-realistic facial transfer have been proposed [Thies et al. 2015; Thies et al. 2016]. At the heart of these methods is a highly-parametrized face prior that models rigid pose, facial identity and expression, skin albedo, and lighting. This model is used to track and parametrize the source and target facial motion in RGB(-D) video streams. Face transfer is then performed by mapping the tracked source motion parameters onto the target model and re-rendering it with the estimated target illumination. In [Thies et al. 2015], source motion parameters are first normalized by aligning the parameter space to the estimated neutral expression prior transfer. A believable mouth interior is modeled with a generic 3D teeth proxy and 2D warping of the oral cavity. Thies et al. [2016] proposed a more sophisticated facial motion mapping function that directly operates in the param-

eter space spanned by the expression basis. A realistic mouth interior is synthesized by retrieving a mouth image from the target sequence, such that the pose, shape, appearance and motion of the mouth agree with that observed in the source sequence. Temporal coherence is enforced by also considering similarity to the last retrieved mouth image, and by blending between the last two retrieved mouth images.

The main disadvantage of the two previous model-based approaches is that they neither capture nor generate person-specific expressions and fine-scale details, which is mandatory in complex video editing tasks. These issues are tackled in Chapters 7–8.

Chapter 4

Image-based Face Capture and Reenactment

Figure 4.1: Reenactment result on a high-quality video (17 s of target footage, 15 s of source footage). *Top:* Example frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites.

This chapter presents a fully-automatic, image-based facial reenactment method that tracks and replaces the face of an actor in an existing target video with that of a user from a source video, while preserving the original target performance (see Figure 4.1). The method and results presented in this chapter are based on [Garrido et al. 2014] and partially on [Garrido et al. 2013].

4.1 Introduction

Face replacement for images and video has been studied extensively (Section 3.5). These techniques substitute a face (or a facial performance) in an existing target image (sequence) with a different face (or performance) from a source image (sequence), and compose a new result that looks realistic. As a particularly challenging case, *video face reenactment* replaces a face in an image sequence, while preserving the gestures and facial expressions of the target actor as much as possible. Since this process requires careful frame-by-frame analysis of the facial performance and the generation of smooth transitions between composites, most existing techniques demand substantial manual interaction.

This chapter presents an entirely image-based method for video face reenactment that is fully automatic and achieves realistic results, even for low-quality video input, such as footage recorded with a webcam. Given an existing *target sequence of an actor* and a self-recorded *source sequence of a user* performing arbitrary face motion, our approach produces a new *reenacted* sequence showing the facial performance of the target actor, but with the face of the user inserted in it. We adhere to the definition of face replacement given by Dale et al. [2011] and only replace the actor's inner face region, while keeping the hair, face outline, and skin color, as well as the background and illumination of the target video. We solve this problem in three steps: First, we track the user and the actor in the source and target sequence using a 2D deformable shape model whose facial features are accurately tracked throughout the sequence. Then, we go over the target sequence and look in the source sequence for frames that are both similar in facial expression and coherent over time. Finally, we adapt the head pose and face shape of the selected source frames to match those of the target, and blend the results in a compositing phase.

Our reenactment solution has several important advantages: 1) Our 2D tracking step is robust under moderate head pose changes and allows a freedom in camera view point. As opposed to existing methods, our system does not require that the user and the target actor share the same pose or face the camera frontally. 2) Our matching step is formulated as an image retrieval task, and, as a result, source and target performances do not have to be similar or of comparable timing. The source sequence is not an exhaustive video database, but a single recording that the user makes of himself going through a short series of non-predetermined facial expressions. Even in the absence of an exact match, our system synthesizes plausible results. 3) Our face transfer step is simple, yet effective, and does not require a 3D face model to map source pose and texture to the target. This saves us the laborious task of generating and tracking a personalized face model, something that is difficult to achieve for existing, pre-recorded footage. 4) None of the above steps needs any manual interaction: Given a source and target video, the reenactment is created automatically.

We further make the following contributions: 1) We present a new optical flow-based correction scheme that uses keyframes to accurately improve the landmark trajectories of a 2D face tracker. 2) We introduce a novel distance metric for matching faces between videos, which combines both appearance and motion information. This allows us to retrieve similar facial expressions, while taking into account temporal continuity. 3) We propose an approach for segmenting the target video into temporal clusters of similar expression, which are compared against the source sequence. This stabilizes matching and assures a more accurate image selection. 4) A final contribution is an image-based warping strategy that preserves facial identity as much as possible. Based on the estimated shape, appearance is transferred by inverse texture warping and image blending.

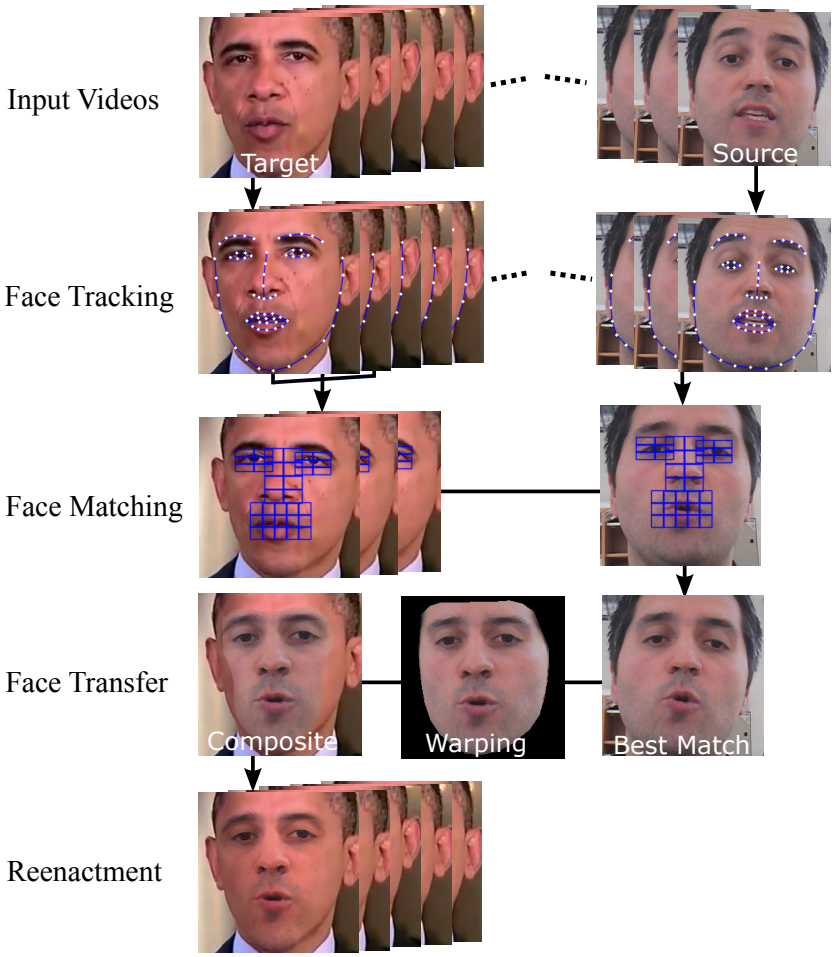


Figure 4.2: Overview of the proposed approach. The system is composed of three main steps: Face tracking (Section 4.3), face matching (Section 4.4), and face transfer (Section 4.5).

4.2 Overview

Our approach takes as input two videos showing facial performances of two different persons: a *source sequence* $\mathcal{S}$ of the user, and a *target sequence* $\mathcal{T}$ of an actor. The goal is to replace the actor's inner face region with that of the user, while preserving the target performance, scene appearance and lighting as faithfully as possible. The result is the *reenactment sequence* $\mathcal{R}$. The source and target video are not assumed to depict the same performance: Reenactments can be produced for different target videos from only a single source video, which is assumed to show the user going through a short series of random facial expressions while facing the camera. The target sequence can be general footage depicting a variety of expressions and head poses.

Our approach consists of three subsequent steps, as illustrated in Figure 4.2:

S1 **Face Tracking (Section 4.3):** A non-rigid face tracking algorithm tracks the user and the actor throughout the videos and provides 2D facial feature points. The landmark trajectories are greatly improved and stabilized using optical flow between automatically selected keyframes.

S2 **Face Matching (Section 4.4):** The appearance of the main facial regions is encoded as a histogram of local binary patterns, and target and source frames are matched by a nearest

neighbor search. This is rendered more stable by dividing the target sequence into temporal chunks of similar appearance and taking into account the motion of the facial landmarks.

S3 **Face Transfer (Section 4.5):** The target head pose is transferred to the selected source frames by warping the accurately tracked facial landmarks. A smooth transition is then created by synthesizing in-between frames and blending the source face into the target sequence using seamless cloning.

4.3 Non-rigid Face Tracking

To track the user and actor in the source and target sequence, respectively, our system utilizes a non-rigid face tracking algorithm proposed by Saragih et al. [2011a], which tracks a sparse set of $m = 66$ consistent landmark locations on the human face, such as the eyes, nose, mouth, and face contour, as shown in Figure 4.3. The approach is an instance of the constrained local model (CLM) [Cristinacce and Cootes 2006] that employs the subspace constrained mean-shift algorithm as an optimization strategy. Specifically, it is based on a 3D point distribution model (PDM), which linearly models non-rigid shape variations around a set of reference landmark locations $\bar{\mathbf{X}}_i \in \mathbb{R}^3$, $\forall i = 1 : m$, and composes them with a global rigid transformation assuming a weak perspective camera model, as follows:

$$\tilde{\mathbf{x}}_i = sPR\left(\bar{\mathbf{X}}_i + \Phi_i \mathbf{q}\right) + \mathbf{t} \quad \text{with} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} . \tag{4.1}$$

Here, $\tilde{\mathbf{x}}_i \in \mathbb{R}^2$, $1 \leq i \leq m$, denotes the estimated 2D location of the i-th facial feature point in the image and P the orthogonal projection matrix. The PDM parameters are the scaling factor s, the 3D rotation matrix R, the 2D translation vector $\mathbf{t}$, and the non-rigid deformation parameters $\mathbf{q} \in \mathbb{R}^z$, where $z = 24$ is the dimension of the PDM model. Furthermore, $\Phi_i \in \mathbb{R}^{3\times z}$ denotes a previously learned submatrix of the basis of variation pertaining to the i-th feature. To find the most likely feature locations, the algorithm first detects the bounding box surrounding the face [Viola and Jones 2004] and calculates a response map for each landmark in the face region by local SVM detectors trained to recognize aligned from misaligned locations. Then, it combines the local detectors in an optimization step that enforces a global prior over the joint motion of the landmarks. Note that the face tracking algorithm utilizes the previously estimated PDM parameters to initialize the optimization in the next frame. Both the trained PDM model and the local feature detectors were provided to us by the authors. It is important to remark that we only use the 2D landmark output $(\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m)$ of the tracker and not the underlying 3D PDM, since the 2D landmark trajectories will in the end be corrected and stabilized in the video, as described below.

4.3.1 Automatic Key Frame Selection

The facial landmarks are prone to noise and inaccuracies, and therefore there may be localization errors in the detected facial landmarks, especially for expressions on which the tracker was not trained. Table 4.1 quantifies this effect by listing the mean distance of the detected landmarks from their manually annotated ground truth locations for a selection of expressions performed by different subjects. This can render the face matching (see Section 4.4) and face transfer (see Section 4.5) less stable. To account for such errors and increase the tracking accuracy, we correct the landmark locations using accurate optical flow between automatically selected *keyframes*, i. e., frames for which the localization of the facial landmarks detected by the face tracker is considered reliable.

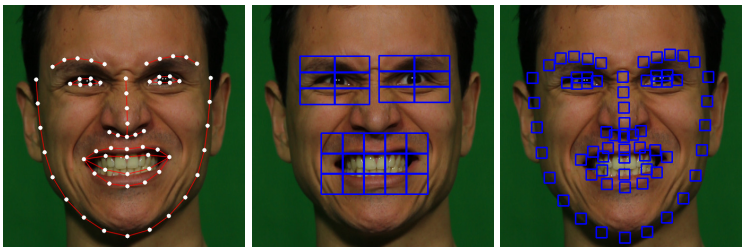


Figure 4.3: Face regions used for generating the LBP descriptors. *Left:* Detected landmarks. *Middle:* Detected mouth and eyes regions, each split into 3×5 and 3×2 tiles, respectively. *Right:* Detected small regions at the $m = 66$ landmark locations.

Appearance Descriptor

Keyframes are selected by comparing the facial appearance of each frame f^t at the timestep t with the appearance of a reference frame f^r that has well localized facial features, for instance, a frame of a neutral pose. Empirically, we learned that the non-rigid face tracking algorithm works fairly well for non-expressive, symmetric facial expressions (i. e., near to neutral expressions). In this thesis, we assume that the first frame in the sequence fulfills these requirements and we choose it as the reference frame.

All frames in the sequence are initially aligned to the first frame using a 2D affine transformation that maps at best the set of detected features onto the reference shape without distorting the face appearance in the image. To extract meaningful facial features in the aligned frames, we consider three rectangular regions of fixed size around the mouth and the eyes, each computed as the bounding box of their corresponding landmark locations in the reference frame f^r. After padding these regions by 25% their size, we split them into several tiles, as shown in Figure 4.3.

As feature descriptor for a region of interest, we choose histograms of local binary patterns (LBPs) [Ahonen et al. 2006; Ojala et al. 2002], which have been found very effective for expression matching and identification tasks [Kemelmacher Shlizerman et al. 2011; Tan and Triggs 2010]. LBPs encode the relative brightness around a pixel by assigning a binary value to each neighboring pixel, depending on whether its intensity is brighter or darker. The result is an integer value between 0 and 2^l for each center pixel, where l is the number of pixels in a circular neighborhood. It is important to remark that the LBP histograms are not quantized, i. e., each representative value of the LBP code is assigned to a single bin in the histogram.

Following [Kemelmacher-Shlizerman et al. 2010], we use a uniform LBP code to achieve simple rotation invariance. This is a particularly important feature as the global alignment described above normally sacrifices registration accuracy to avoid distorting the appearance of the face, especially for out-of-plane head rotations. Uniform LBP codes assign an own label to every binary combination for which the number of bitwise transitions between 0 and 1 (or vice versa) is at most two when the bit pattern is traversed circularly, and a single label for all other combinations. For a neighborhood size of $l = 8$, this results in an LBP histogram h of 59 bins [Ojala et al. 2002] for each tile. Empirically, we found that uniform codes lack in discerning power to recognize expressions from a wider set other than the distinctive neutral, sadness, happiness and anger expressions. Even though this is not crucial when selecting keyframes, uniform codes are insufficient for accurately matching expressions across individuals (see Section 4.4). Hence, to increase the discriminating power of appearance matching at a finer scale of detail, we additionally compute a standard LBP histogram for a neighborhood size of $l = 4$, thereby extending h to $59 + 2^4 = 75$ bins for each tile. This gives a good

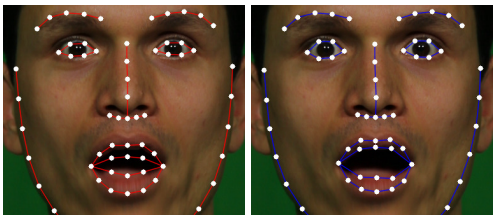


Figure 4.4: Position of facial landmarks before correction (left) and after correction (right). Note that the correction clearly improves the tracking of certain features, such as the mouth region and the eyes.

trade-off between complexity and matching accuracy. Finally, by concatenating the histograms for all J tiles that make up a region of interest, an LBP feature descriptor $H=(h_1,\ldots,h_J)$ for the whole region is created.

Appearance Matching

In a first pass, an initial set of keyframes is selected as those frames in the sequence that are closest to the reference expression according to the following distance metric:

$$d_{\text{app}}(f^r,f^t)=\sum_{j=1}^{3} d_{\chi^2}\left(H_j(f^r),H_j(f^t)\right) \quad , \tag{4.2}$$

where d_{χ^2} is the normalized chi-squared distance between two histograms and H_j the appearance descriptor for the eyes and mouth regions. The amount of initial keyframes is chosen as $x\%$ of the sequence length, which can be thought of as a probability estimate of finding the reference expression in the video and at the same time corresponds to an average inter-key-frame distance.

In a second pass, we select clips between consecutive keyframes with a length of more than y frames and divide these by adding more keyframes. These in-between keyframes are selected in the same way using the distance metric Equation 4.2, but this time we use a sparse appearance descriptor H for a small squared region around each of the $m=66$ detected facial landmarks (see Figure 4.3). The size of each region was set to 10% the size of bounding box representing the inner part of the face (eyebrows and face contour). Unlike the initial keyframes, in-between keyframes may not have the same expression as the reference, since we only seek similar texture patterns around facial feature points and not within whole facial regions. The division threshold of y frames is chosen in such a way it limits the inherent drift by optic flow (see Section 4.3.2) over longer clips.

For the results presented in this chapter as well as in Chapter 5, $x=2.5\%$ and $y=40$. These values were found empirically. In some experiments conducted on the sequences presented in Chapter 5, the resulting average distance between keyframes was 22, with an average maximum of almost 90. Note that such sequences exhibit fast and expressive facial motions, and the first frame is assumed to be at rest pose. We also remark that videos containing substantially different characteristics may require further adjustment of the empirical values reported above.

4.3.2 Optical Flow-based Feature Correction

If we assume that we have a keyframe at time $t=T$, we compute the landmark locations at times $t>T$, as follows:

$$\mathbf{x}_i^t=\lambda_i\,\tilde{\mathbf{x}}_i^t+(1-\lambda_i)\,\mathbf{x}_{\text{o},i}^t \quad \text{for} \quad 1\le i\le m \quad , \tag{4.3}$$

Table 4.1: Keyframe-based landmark correction: Mean distance (in pixels) of the 66 tracked landmarks to their manually annotated ground truth location for a selection of expressions from the sequences shown in the experiments of Section 4.6.1 and Section 5.7.1.

Sequence	Feature Tracking	Key Frame Correction
11 expressions of seq. 1 (Figure 4.14)	5.38 ± 1.47	3.83 ± 1.05
11 expressions of seq. 2 (Figure 5.8)	6.72 ± 1.44	4.60 ± 0.70
10 expressions of seq. 3 (Figure 5.9)	6.36 ± 1.65	4.13 ± 0.88
Overall	6.15 ± 1.52	4.19 ± 0.88
Overall, only mouth and eyes	7.24 ± 2.22	4.35 ± 1.46

where $0 \leq \lambda_i \leq 1$ is a weighting factor. In this expression, $\tilde{\mathbf{x}}^t \in \mathbb{R}^{2\times m}$ are the facial landmark positions (see Equation 4.1) obtained by the *facial feature tracker* at time t, while $\mathbf{x}_o^t \in \mathbb{R}^{2\times m}$ are the locations estimated by *optical flow*:

$$\mathbf{x}_o^t = \mathbf{x}^T + \sum_{T \leq q < t} \mathbf{w}^q \ . \tag{4.4}$$

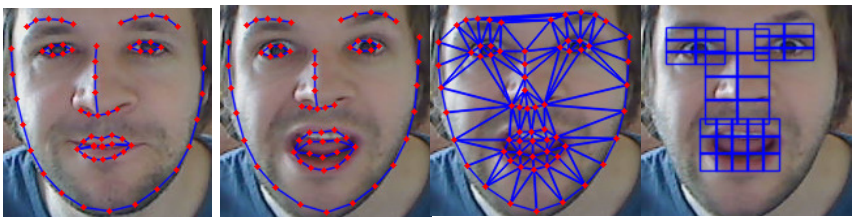
Here, $\mathbf{x}^T$ denotes the landmark positions in the keyframe f^T and $\mathbf{w}^t$ is the forward optical flow vector field from t to $t+1$ in $\mathbf{x}_o^t$. Optical flow is estimated in a variational framework by minimizing an energy consisting of a data term with brightness and gradient constancy assumption, and a structure-aware smoothness term, as proposed in [Valgaerts et al. 2012b]. To further stabilize the tracking of $\mathbf{x}_o^t$ between keyframes, we also compute the backward optical flow from $t+1$ to t and use it to back-trace the landmark position from the next keyframe. To be more precise, the optical flow-based correction strategy is performed in both directions, where the influence of the forward and backward optical flows is varied smoothly over time, with the forward (backward) flow having more weight near the previous (next) keyframe, respectively. This avoids an accumulation of drift errors and also ensures smooth landmark trajectories at keyframes. A related keyframe approach for dense tracking was adopted by Beeler et al. [2011].

Improvements in feature point location after optical flow-based correction are clearly noticeable for very expressive regions, such as the mouth and the eyes in Figure 4.4. Table 4.1 further shows that overall the localization of facial feature points improves after our correction step.

To improve the smoothness of the landmark trajectories even further, we do not use the estimated optical flow value at the exact landmark location $\mathbf{x}_i$, but assign a weighted average of the flow in a circular neighborhood around $\mathbf{x}_i$. This neighborhood of size $r \cdot p$ is built by distributing p points evenly on circles with radial distances of $1, 2, \ldots, r$ from $\mathbf{x}_i$. In our experiments we choose $r=2$ and $p=8$, and weigh the flow values by a normalized Gaussian centered at $\mathbf{x}_i$.

4.4 Face Matching

A central part of our reenactment system is matching the source and target faces under differences in head pose. Here, we find a trade-off between exact expression matching, and temporal stability and coherence. The tracking step of the previous section provides us with accurate facial landmarks that coarsely represent the face shape. Instead of comparing shapes directly, we match faces based on appearance and landmark motion, depicting the facial expression and its rate of change, respectively. Another contribution of the matching step is a temporal clustering approach that renders the matching process more stable.



(a) (b)

Figure 4.5: Image alignment and feature extraction. (a) Reference frame used for alignment (with its respective estimated landmarks). (b) Expressive face aligned to the reference. *Left to right:* Estimated landmarks, triangulation and detected regions of interest. The mouth, eyes and nose regions are split into 3×5, 3×2 and 4×2 tiles, respectively.

4.4.1 Image Alignment and Feature Extraction

Before extracting meaningful facial features, the source and target frames are first aligned to a common reference frame, as shown in Figure 4.5 (a). For this purpose, we choose the first frame in the source sequence, which is assumed to depict the user at rest pose. Unlike methods that align source and target using a morphable 3D model [Kemelmacher-Shlizerman et al. 2010], we compute a 2D affine transformation for each frame that optimally maps the set of detected landmarks onto the reference shape, as described in Section 4.3.1. Since this transformation is global, it does not change the expression in the aligned frames. This alignment is only necessary for the temporal clustering and frame selection of Section 4.4.2, but is not applied for the subsequent steps of the system.

To be able to recognize similar expressions between individuals at a finer level, we consider in this case four regions of interest of fixed size: The eyes (left and right, separately), the mouth, and the nose, as shown in Figure 4.5 (b). Since source and target frames are aligned, these regions are computed only once in the reference source frame. Note that the nose region partially contains nasolabial features as well as the area in which frown lines form, which together are important to distinguish, for instance, a smile from anger. The LBP descriptors are computed in the same way as described in Section 4.3.1.

4.4.2 Temporal Clustering and Frame Selection

Matching source and target frames directly may lead to abrupt frame-to-frame expression changes in the reenactment. The reasons for this are: 1) We experienced a sensitivity of LBP feature descriptors w. r. t. the detected regions of interest, which can result in slightly different selection of source frames for similar target expressions (comparable effects were reported by Li et al. [2012]). 2) The source sequence is sparse and may not contain an exact match for each target expression. 3) There is no temporal consistency in the image selection. To overcome these shortcomings, we stabilize the matching process by a temporal clustering approach, which finds the source frame that is most similar to a small section of target frames. Additionally, we enforce temporal continuity by extending the appearance metric with a motion similarity term, which takes into account the change in expression.

Temporal Clustering

To stabilize the selection of source frames, we divide the target sequence into consecutive sections of variable length based on expression and appearance similarity, and then look for the source frame that best matches a whole target section. To measure the similarity between two consecutive target frames $f_{\mathcal{T}}^t, f_{\mathcal{T}}^{t+1} \in \mathcal{T}$, we compute the *appearance distance*

$$d_{\text{app}}(f_{\mathcal{T}}^t, f_{\mathcal{T}}^{t+1}) = \sum_{j=1}^{4} w_j \, d_{\chi^2}\left(H_j(f_{\mathcal{T}}^t), H_j(f_{\mathcal{T}}^{t+1})\right) \quad , \tag{4.5}$$

where $H_j(f)$ is the LBP feature descriptor for the j-th of the four regions of interest in f, w_j an accompanying weight, and d_{χ^2} the chi-squared distance. The weights for mouth, eyes and nose regions were experimentally set to 0.6, 0.15 and 0.1, respectively.

The proposed clustering approach is related to hierarchical agglomerative clustering methods, but it is explicitly designed to preserve temporal continuity, i. e., it only merges clusters that are consecutive in time, thereby preserving the order of the target frames. Similar to agglomerative-based approaches, our algorithm proceeds hierarchically, as follows: Assuming that each frame is initially a separate cluster, each subsequent iteration joins the two consecutive clusters that are closest according to the metric in Equation 4.5. As a linkage criterion, the appearance distance between two consecutive clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ is defined as the average of the pairwise distances d_{app} between all frames in $\mathcal{C}_1$ and all frames in $\mathcal{C}_2$. The two closest consecutive clusters are finally merged if 1) they only contain a single frame or 2) the variance of d_{app} within the merged cluster is smaller than the maximum of the variances within the separate clusters. The last criterion keeps the frames within a cluster as similar as possible, and once it is not met, the algorithm terminates. An advantage of our clustering approach is that it is parameter-free, thus no tuning is required. The result is a sequence of target sections $\mathcal{C}^k$, with k an index running in temporal direction over the number of clusters.

We observed that the length of a cluster $\mathcal{C}$ varies inversely proportionally to the change in expression and the timing of speech within $\mathcal{C}$. An analysis of the number of detected clusters and their lengths can be seen in Figure 4.6. This figure shows a plot of the distance metric d_{app} between two consecutive frames for 32 frames of the target sequence depicted in Figure 4.15. The target clusters that are computed by our temporal clustering approach are drawn as red lines below the graph, while isolated frames and boundary frames are indicated by green squares. The values of the distance metric d_{app} are drawn as red circles enclosed by the frames between which it measures the similarity.

As one would expect, consecutive frames are merged into a cluster if the value of d_{app} is low. If d_{app} remains low for an extended number of consecutive frames, a large cluster is formed, such as the one spanning frames 48 to 52. Peaks in the graph indicate dissimilar frames and these typically form cluster boundaries or isolated frames. Note that the graph is dynamic and changes as the algorithm proceeds since the value of d_{app} between consecutive clusters changes as more clusters are formed (difficult to visualize).

To illustrate the similarity in appearance of frames within the same cluster, we display the boundary frames of the cluster spanning frames 38 to 41 at the bottom left, the cluster spanning frames 48 to 52 in the top middle, and the cluster spanning frames 53 to 55 at the bottom right of the figure. The two examples of isolated frames shown at the top left and right side lie outside of a cluster and differ in appearance from those within the neighboring clusters. It can be concluded that the length of a cluster roughly varies inversely proportionally to the change in expression and the timing of speech within the cluster. The maximum and average cluster length as well as the total number of clusters

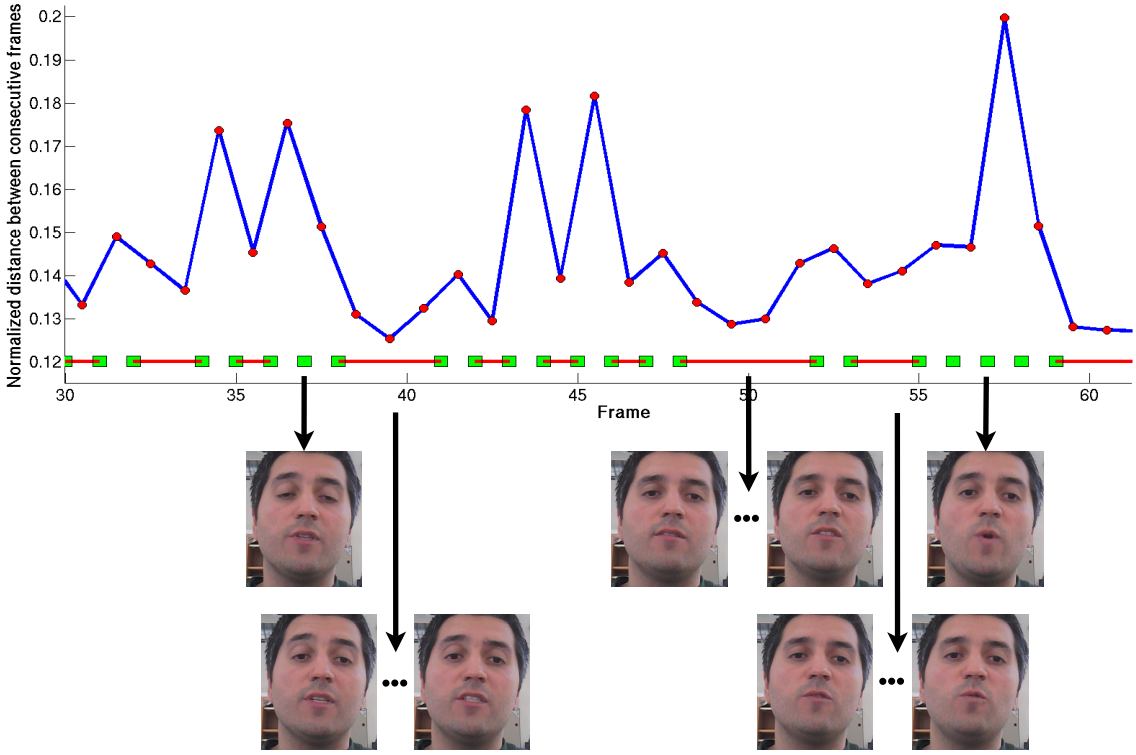


Figure 4.6: Plot of the distance metric d_{app} and the resulting clusters obtained in the target sequence by the proposed temporal clustering approach.

Table 4.2: The mean (mean size) and maximum (max. size) cluster size, and the total number of clusters (num. clusters) computed for the sequences shown in Section 4.6.

Sequence	num. frames	num. clusters	mean size	max. size
Figure 4.12	231	86	2.7	5
Figure 4.13	524	196	2.7	6
Figure 4.15	200	59	3.5	15
Figure 4.9	446	155	2.9	8
Figure 4.10	566	215	2.6	6
Figure 4.11	352	136	2.6	4
Figure 4.16	319	128	2.5	5
Figure 4.1	533	191	2.8	9

computed for the target sequences of the figures shown in Section 4.6.1 are given in Table 4.2. For the results presented here, we enforced the minimum cluster size to be 2, which generally leads to smoother animations for sequences with many isolated frames. Enforcing this is easily done by adding isolated frames to the left or right cluster, depending on which one is closest in d_{app}.

Frame Selection

To select a source frame $f_s^k \in \mathcal{S}$ that matches a target section $\mathcal{C}^k$, we compute an aggregated similarity metric over all target frames in a cluster:

$$d(\mathcal{C}, f_s) = \sum_{f_\mathcal{T} \in \mathcal{C}} d_{\text{app}}(f_\mathcal{T}, f_s) + \tau \, d_{\text{mot}}(\mathbf{v}_C, \mathbf{v}_s) \ . \tag{4.6}$$

Here, $d_{\text{app}}(f_1, f_2)$ is the appearance distance defined in Equation 4.5 and $d_{\text{mot}}(\mathbf{v}_1, \mathbf{v}_2)$ a *motion distance* that measures the similarity between two vector fields $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{2\times m}$. The vector field $\mathbf{v}_C$ describes the motion of the m facial landmarks between two consecutive clusters. The motion of the i-th landmark $\mathbf{v}_{C,i}$ is computed as the difference of its average positions in the current cluster $\mathcal{C}^k$ and the previous cluster $\mathcal{C}^{k-1}$. The vector field $\mathbf{v}_s$ describes the motion of the m facial landmarks between two consecutively selected source frames, i. e., for the i-th landmark, $\mathbf{v}_{s,i}$ is the difference of its position in f_s^k and f_s^{k-1}. Note that $\mathbf{v}_C$ and $\mathbf{v}_s$ are computed for normalized landmark locations in the aligned source and target frames. The motion distance d_{mot} is defined as

$$d_{\text{mot}}(\mathbf{v}_C, \mathbf{v}_s) = 1 - \frac{1}{3} \sum_{j=1}^{3} \exp\big(-d_j(\mathbf{v}_C, \mathbf{v}_s)\big) \ , \tag{4.7}$$

$$d_1 = \frac{1}{m} \sum_i \|\mathbf{v}_{C,i} - \mathbf{v}_{s,i}\|_2 \ , \tag{4.8}$$

$$d_2 = \frac{1}{m} \sum_i \left(1 - \frac{\mathbf{v}_{C,i} \cdot \mathbf{v}_{s,i}}{\|\mathbf{v}_{C,i}\|_2 \|\mathbf{v}_{s,i}\|_2}\right) \ , \tag{4.9}$$

$$d_3 = \frac{1}{m} \sum_i \big| \|\mathbf{v}_{C,i}\|_2 - \|\mathbf{v}_{s,i}\|_2 \big| \ , \tag{4.10}$$

where d_1 measures the Euclidean distance, d_2 the angular distance, and d_3 the difference in magnitude between the motion fields $\mathbf{v}_C$ and $\mathbf{v}_s$. The motion distance d_{mot} therefore measures how similar the change in expression in the selected source frames is compared to the change in expression between target clusters. It is important to understand that consecutively selected frames f_s^{k-1} and f_s^k do not have to be consecutive in the original source sequence $\mathcal{S}$. The matching metric in Equation 4.6 is thus suitable for source and target sequences that have an entirely different timing and speed. We remark that both the aggregated appearance distance and motion distance are normalized to $[0, 1]$ and the weighting factor τ was set to 0.8 for all experiments.

Given f_s^{k-1}, the source frame with the minimal total distance $d(\mathcal{C}^k, f_s)$ over all $f_s \in \mathcal{S}$, is chosen as the best match f_s^k and assigned to the central timestamp of $\mathcal{C}^k$. If $\mathcal{C}^k$ consists of a single frame, f_s^k is assigned to this timestamp.

4.5 Face Transfer

After selecting the best representative source frames, we transfer the face of the user to the corresponding target frames and create the final composite. First, we employ a 2D warping approach that combines global and local transformations to produce a natural shape deformation of the user's face that matches the actor in the target sequence. The estimated shape is then utilized to transfer the user's appearance and synthesize a compelling transition.

4.5.1 Shape and Appearance Transfer

While only methods relying on complex 3D face models can handle large differences in head pose between source and target [Dale et al. 2011], we present a simple, yet effective, image-based strategy that succeeds in such cases. Inspired by work on non-rigid template fitting [Blanz et al. 2004; Weise et al. 2009], we formulate face transfer as a deformable 2D shape registration that finds a user shape and pose that best correspond to the shape and pose of the actor, while preserving the user's identity as much as possible.

Shape Transfer

For each target frame $f_{\mathcal{T}}^t \in \mathcal{T}$, we want to estimate the m 2D landmark locations $\left(\mathbf{x}_{\mathcal{R},1}^t, ..., \mathbf{x}_{\mathcal{R},m}^t\right)$ of the user's face in the reenactment sequence $\mathcal{R}$. To achieve this, we propose a warping energy composed of two terms: a non-rigid term and an affine term. The non-rigid term penalizes deviations from the target shape:

$$E_{\text{nr}} = \sum_{i=1}^{m} \left\| \mathbf{x}_{\mathcal{R},i}^t - \left(\alpha_1 \mathbf{x}_{\mathcal{T},i}^{t-1} + \alpha_2 \mathbf{x}_{\mathcal{T},i}^t + \alpha_3 \mathbf{x}_{\mathcal{T},i}^{t+1} \right) \right\|^2 , \tag{4.11}$$

where $\mathbf{x}_{\mathcal{T},i}^t$ denotes the i-th landmark in the target frame at time t and α_j, $\sum_j \alpha_j = 1$, are normalized weights (0.1, 0.8 and 0.1 in our experiments). The affine term penalizes deviations from the selected source shape:

$$E_{\text{r}} = \sum_{i=1}^{m} \left\| \mathbf{x}_{\mathcal{R},i}^t - \left(\beta_1 M^{k-1} \mathbf{x}_{S,i}^{k-1} + \beta_2 M^k \mathbf{x}_{S,i}^k \right) \right\|^2 , \tag{4.12}$$

where $\mathbf{x}_{S,i}^{k-1}$ (resp. $\mathbf{x}_{S,i}^k$) is the i-th landmark in the selected source frame immediately preceding (resp. following) the current timestamp t, and M^k a global affine transformation matrix that optimally aligns $\mathbf{x}_S^k$ and $\mathbf{x}_{\mathcal{T}}^c$, with c the central timestamp in the k-th cluster. As the selected source frames are only assigned to the central timestamp of a temporal cluster, no selected source shape may correspond to the current target frame $f_{\mathcal{T}}^t$, so this term effectively interpolates between the closest selected source shapes, thereby preserving the user's identity. The weights β_j, $\sum_j \beta_j = 1$, depend linearly on the relative distance from t to the central timestamps of $\mathcal{C}^{k-1}$ and $\mathcal{C}^k$, being 0 or 1 if t coincides with one of the cluster centers. Combining the two terms together with their corresponding weights w_{nr} and w_{r}, yields the total energy

$$E_{\text{tot}}(\mathbf{x}_{\mathcal{R},i}^t) = w_{\text{nr}} E_{\text{nr}} + w_{\text{r}} E_{\text{r}} , \tag{4.13}$$

where $w_{\text{nr}} + w_{\text{r}} = 1$. A closed-form solution to Equation 4.13 for the optimal landmark locations $\left(\mathbf{x}_{\mathcal{R},1}^t, ..., \mathbf{x}_{\mathcal{R},m}^t\right)$ exists. Note that the values of the trade-off weights $w_{\text{nr}} \in \{0.55, 0.65\}$, $w_{\text{r}} \in \{0.45, 0.35\}$ were found empirically and mainly selected based on the amount of out-of-plane head rotation that exists in the target sequence (larger out-of-plane head rotation angles imply higher face deformation, and therefore higher influence of the non-rigid term). Please refer to Section 4.6 to see the values assigned to each sequence.

Appearance Transfer

Once we have the optimal shape of the face in the reenactment sequence, we transfer the appearance of the selected source frames by inverse-warping the corresponding source texture using a triangulation of the landmark points (see Figure 4.5 (b)), as proposed in [Saragih et al. 2011a]. For the

Figure 4.7: Comparison of warping approaches. *Left:* Selected user frame. *Right:* Target pose. *Middle left to right:* Non-rigid warping (Equation 4.11), affine warping (Equation 4.12), and our approach (Equation 4.13). Note that non-rigid warping distorts eyes and mouth, while affine warping fails to find a correct view deformation.

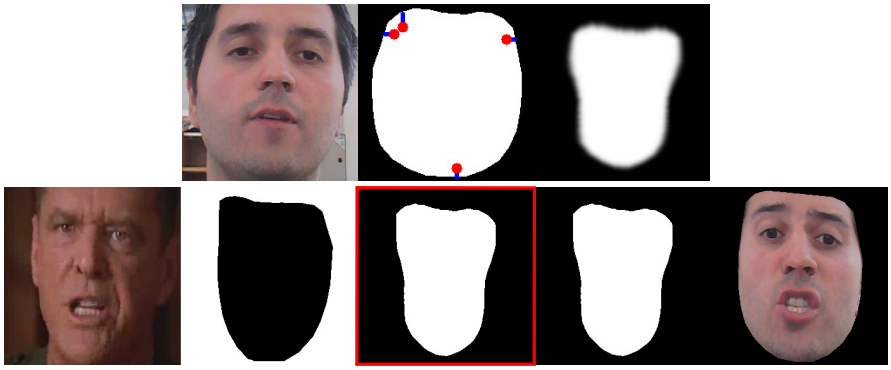


Figure 4.8: Seam generation. *Top:* User at rest pose, source mask with landmarks closest to the boundary in red, and eroded mask. *Bottom left:* Target frame and mask. *Bottom right:* Transferred source frame and mask. *Bottom middle:* Final blending seam obtained by intersecting source and target mask.

in-between frames, we create a smooth transition in appearance by interpolating the texture from the closest selected source frames using the same triangulation of the landmarks.

Note that a shape and appearance transfer as described here are generally not possible with conventional warping approaches, such as global non-rigid warping and global affine warping, as shown in Figure 4.7. The former creates unrealistic distortions in texture since it fits the source shape exactly to the target shape, while the latter may fail under strong perspective views and create odd deformations whenever the source and target shape do not agree.

Compositing

Having transferred the source face to the target sequence, we produce a convincing composite, where the main facial source features, represented by the eyes, nose, mouth, and chin are seamlessly implanted onto the target actor. The lighting of the target sequence, and the skin appearance and hair of the target actor, should be preserved. For this purpose, we use Poisson seamless cloning [Pérez et al. 2003]. We create a tight binary mask for the source sequence containing the main facial features of the user at rest, such as eyes, mouth, nose and eyebrows. We then perform an erosion with a Gaussian structuring element that is constrained by the landmark locations in the

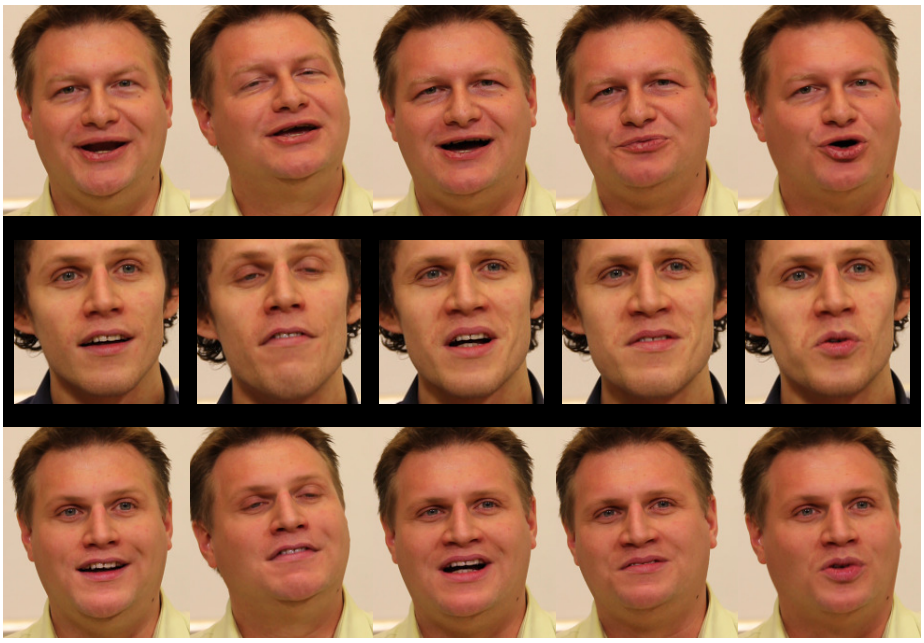


Figure 4.9: Existing high-quality video (17 s of target footage, 10 s of source footage). *Top:* Example frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.65$, $w_{\text{r}} = 0.35$.

facial features. Thresholding this mask gives us a seam for blending (see Figure 4.8, top).

To obtain a seam for each frame in the reenactment, the precomputed source mask is transferred by inverse-warping (see Section 4.5.1). We prevent the seam from running outside the target face by intersecting it with a mask containing the main facial features of the target actor (see Figure 4.8, bottom). For increased insensitivity to the source illumination, we transform the source and target frames into the perception-based color space of [Chong et al. 2008] before performing Poisson blending [Pérez et al. 2003]. The blended image is converted back to RGB space, resulting in the final composite (see Figure 4.2). To avoid artifacts across the seam, we blend the boundary pixels using a Gaussian with a standard deviation of 9 pixels.

4.6 Experiments

We evaluate our method on two types of data: We use videos that were pre-recorded in a studio with an SLR camera at 25 fps to demonstrate the reenactment quality on existing high-quality footage. We also reenact faces in videos taken from the Internet using a random performance of a user captured with a webcam. This demonstrates our system's ease of use and its applicability to online content. Our system was implemented in C++ and tested on a 3.4 GHz Intel®Core™ i5 processor with 16GB RAM. As the results shown below are viewed best as video, we encourage the reader to watch the supplemental video at the project website[1].

[1] http://gvv.mpi-inf.mpg.de/projects/FaceReenactment/

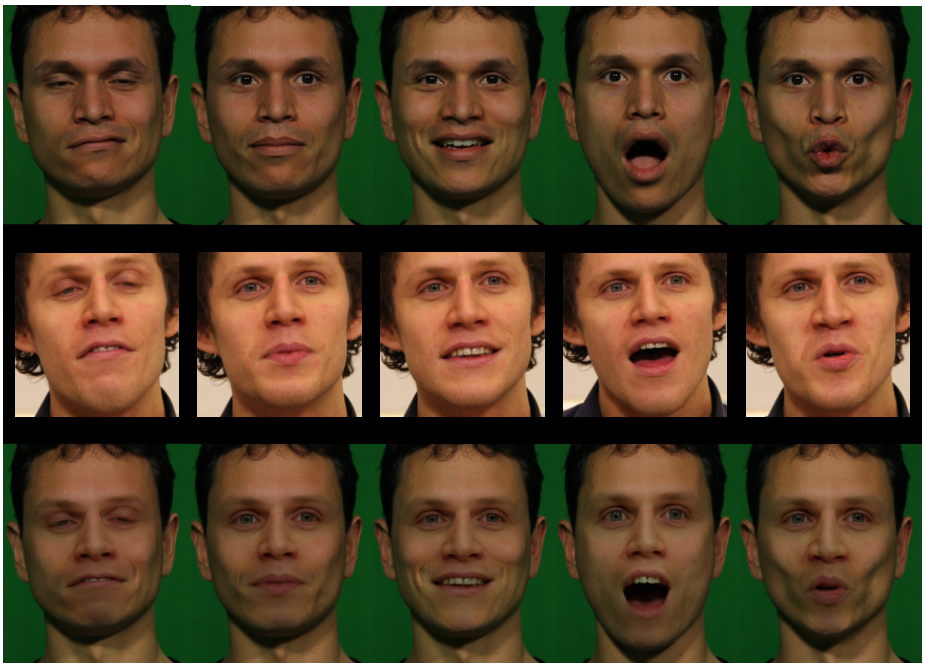


Figure 4.10: Existing high-quality video (22 s of target footage, 10 s of source footage). *Top:* Example frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.65$, $w_{\text{r}} = 0.35$.

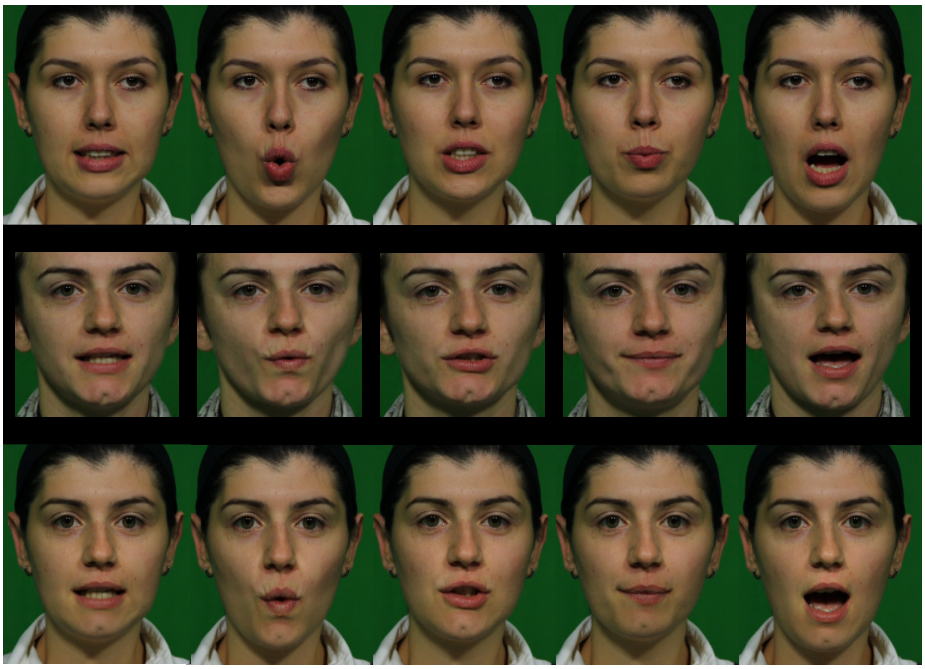


Figure 4.11: Existing high-quality video (14 s of target footage, 10 s of source footage). *Top:* Example frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.55$, $w_{\text{r}} = 0.45$.

Figure 4.12: Low-quality video from the Internet (8 s of target footage, 10 s of source footage). Excerpt from "A Few Good Men" (http://youtu.be/5j2F4VcBmeo). *Top:* Frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.55$, $w_{\text{r}} = 0.45$.

4.6.1 Results

Existing Video We recorded three male and two female users performing random facial gestures and speech under similar ambient lighting to simulate existing high-quality HD footage. As source sequences, we selected from the recordings a snippet of about 10 s showing one of the males and one of the females, and utilized the recordings of the other subjects as target. Figure 4.9, Figure 4.10, Figure 4.11 show three reenactment results of 17, 22 and 14 s. Note that our system is able to reproduce the target performance in a convincing way, even when head motion, expression, timing, and speech of user and actor differ substantially. Computation time for the face tracking step was about 4 s per frame, while the combined face matching and face transfer took around $4 \sim 6$ min for processing the whole sequence. Please refer to the supplementary video at the project website to appreciate the temporal quality of these and other additional results.

Low-Quality Internet Video Figure 4.12 and Figure 4.13 show results for two target videos downloaded from the Internet. The user recorded himself with a standard webcam (20 fps, 640×480) for 10 s, and the reenactments were produced for subsequences of 8 s and 18 s. Both target videos exhibit speech, head pose, lighting and resolution that differ from the recorded source sequence. Our system nevertheless produces plausible animations, even in the presence of quite some head motion, such as in the Obama sequence (see Figure 4.13). Here, face matching and face transfer took between 4 and 7 min for processing the whole sequence.

Figure 4.13: Low-quality video from the Internet (18 s of target footage, 10 s of source footage). President Obama's speech (http://youtu.be/qxtydXN3f1U). *Top:* Frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.65$, $w_{\text{r}} = 0.35$.

4.6.2 Validations

User Study We evaluated the different contributions of our approach by comparing our full reenactment system with (1) a simplified system that does not include the temporal clustering approach proposed in Section 4.4.2 (i. e., a straightforward frame-by-frame matching) and (2) a basic system that does not include temporal clustering, nor does it consider the motion distance defined in Equation 4.7 (i. e., a pure frame-by-frame matching that does not enforce temporally-coherent motion of landmarks). To this end, we performed a user study with 32 participants. The participants were asked to rate reenactment results for two low-quality (LQ) web videos and four existing high-quality (HQ) videos with respect to the original target performance in terms of mimicking fidelity, temporal consistency and visual artifacts on a scale from 1 (not good) to 5 (good). The study was conducted as a web page survey and sent around to a general audience of non-experts that were not aware of the techniques employed to generate the reenactments. Table 4.3 shows the average rating for the six reenactment results. From these results, we conclude that our full system (3.25 average over all sequences) outperforms systems without temporal clustering (2.92), and additionally without combined appearance and motion distance (1.48). These results are statistically significant as the ANOVA p-value for each sequence was on average below 10^{-5}. Overall, the scores for the HQ sequences were higher than for the LQ web videos. These scores should not be directly compared to those reported by Li et al. [2012], since we evaluated different methods and asked different questions.

Self-reenactment Figure 4.14 illustrates a particular example of a self-reenactment, i. e., a reenactment result obtained by taking the same video sequence, both as source and target. Ideally, such a result should be identical to the input videos, and it can be used to test the performance of a reenactment system, for instance, by examining visual artifacts that are introduced in the original

Table 4.3: Results of a user study with 32 participants and six of our reenactment results. The scores listed below denote the average of a rating between 1 (not good) and 5 (good) w. r. t. the original target performance in terms of mimicking fidelity, temporal consistency, and visual artifacts. The results used in the study are the ones referred to by the figure number. Note that (1) means the full system without temporal clustering, and (2) the full system without temporal clustering and motion distance.

Sequence	LQ video		HQ video			
	Figure 4.12	Figure 4.13	Figure 4.9	Figure 4.10	Figure 4.11	Figure 4.16
Full system	**2.5**	**3.56**	**3.19**	**3.00**	**3.38**	**3.81**
(1)	2.09	2.84	3.16	2.72	3.06	3.47
(2)	1.34	1.34	1.41	1.16	1.50	2.16

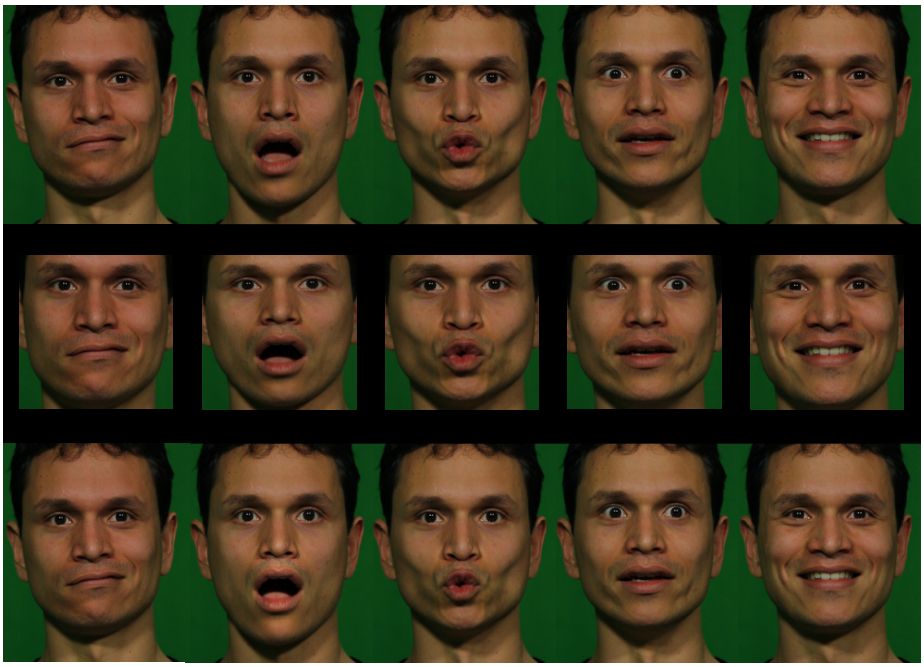


Figure 4.14: Self-reenactment result computed on existing high-quality video (22 s of target and source footage). *Top:* Example frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.55$, $w_{\text{r}} = 0.45$.

sequence.

The self-reenactment shown in Figure 4.14 is almost indistinguishable in appearance and expression from the source and target video. If we define a mismatch as a source frame that is assigned to a target cluster in which it is not contained (source and target are the same video), our system produced 36 mismatches on a total of 214 clusters (22 s of video). The first two columns in Figure 4.14 show two of such mismatches, where a cluster that appears earlier in the sequence was matched to a later frame. However, as it can be observed, these mismatches are very similar in appearance to the frames in the target clusters and the final reenactment is visually close to a perfect frame-by-frame synthesis of the true target sequence. This similarity is confirmed by an average PSNR of 41 dB over 566 frames, with a minimum of 33 dB. Figure 4.15 shows a self-reenactment of a low-quality 10 s webcam sequence. We obtained 1 mismatch on 59 computed clusters.

Figure 4.15: Self-reenactment result computed on low-quality video (10 s of target and source footage). *Top:* Frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.55$, $w_{\text{r}} = 0.45$.

Also for the case where source and target depict the same person under similar conditions, the reenactment resembles the target sequence closely. An example is shown in Figure 4.16, where the source and target sequence are different excerpts taken from a 100 s recording of the same person. Both excerpts were selected arbitrarily without considering possible similarities in the actor's performance. The figure and the supplementary video at the project website show that the final reenactment is very convincing and realistic, a result that was also highly appreciated in the user study, see Table 4.3 (last column).

Length of the Source Video and Reenactment Quality To demonstrate the influence of the source data size on the reenactment quality, we repeated our experiments for successively shorter source sequences, i. e., by taking the first 50%, 25%, and 12.5% of the source material. The supplementary video at the project website shows such a test for the self-reenactment of Figure 4.14. We conclude that a small amount of source frames may lead to unnatural results, with static expressions that appear to be stuck on a moving face (due to certain frames being selected repeatedly and warped to less likely locations in the target face). Longer source sequences clearly result in more realistically reenacted expressions and fewer abrupt transitions, since the newly included source frames cover more of the expressions in the target sequence. However, for many of our examples, the deterioration in reenactment quality with increasingly shorter source sequences was not as pronounced. This shows that we can even produce plausible results for a small set of source frames.

A near-perfect reenactment could be achieved for any target sequence by using a huge amount of meticulously preselected source frames that span a large dictionary of possible expressions. However, such results would strongly depend on the choice of database, while the aim in this chapter is to demonstrate that our method works for videos containing arbitrary facial expressions.

Figure 4.16: Reenactment of the same person under similar conditions in existing high-quality video (12 s of target footage, 14 s of source footage). *Top:* Example frames from the target sequence. *Middle:* Corresponding selected source frames. *Bottom:* Final composites. Chosen weights in Equation 4.13: $w_{\text{nr}} = 0.55$, $w_{\text{r}} = 0.45$.

Comparison with Dale et al. We also compared our fully automatic reenactment system with the semi-automatic face replacement system of Dale et al. [2011] on data provided by the authors. The source and target sequences depict two different subjects reciting the same poem. Our reenactment result is shown in Figure 4.1 and in a side-by-side comparison with the result of Dale et al. in the supplementary video at the project website, demonstrating that they are very close in visual quality. For this result, we selected the following weights in Equation 4.13: $w_{\text{nr}} = 0.65$, $w_{\text{r}} = 0.35$.

Note that a direct frame-by-frame comparison of both results is not meaningful since the method of Dale et al. transfers the source face, including the complete source performance, while our method only transfers the source face, but preserves the target performance. Because source and target performance for this example are slightly different (due to the poem being recited by two different actors), both results differ visually as well. Strictly speaking, the result of Dale et al. is not a "reenactment": Their method warps the target timeline to match that of the source performance and transfers the source face, including its complete performance, which may be considered an easier task since it inherently ensures temporal continuity in the final composite.

4.7 Discussion and Limitations

Despite differences in speech, timing and lighting, the proposed approach creates credible animations, provided that the lighting remains constant or changes globally. Local lighting variations can lead to wrong color propagation across the seam, and can produce flicker and less realistic reenactments. Ghosting artifacts may also appear in the mouth region, stemming from blending and

temporal inconsistencies. In Chapter 6, we tackle most of these problems by making the compositing step more robust to lighting changes, and by replacing the proposed image-based approach with a model-based approach (see Chapter 5) that allows us to drive the mouth separately.

Although our aim is to closely reproduce the facial expressions in the target sequence, the obtained reenactment results can differ from the original performance due to the lack of matching expressions in the source sequence, or the limited precision of the matching metric proposed in Section 4.4.2. Even for source and target performances under perfect matching conditions, the proposed image-based method will still preserve person-specific nuances and subtle specialties of the source expressions, which not only differ in detail from the target expressions, but also between individual users of the system.

All the results shown in this chapter, as well as in subsequent chapters, rely on accurate tracking of 2D facial landmarks to detect the face and relevant facial features. In Section 4.3.1, we have shown an efficient keyframe-based method that improves the landmark trajectories of an off-the-shelf face tracking algorithm [Saragih et al. 2011a]. The landmark correction strategy, however, assumes that the reference frame exists (or is similar to some frames) in the video. This assumption may fail if the tracked sequences contain expressive faces that greatly differ from the reference.

Unlike previous image-based approaches or model-assisted image-based methods for face replacement/reenactment (see Section 3.5), the proposed method succeeds in transferring face motion and appearance for target sequences exhibiting quite some head motion, including moderate out-of-plane head rotations. It is even robust if source and target head poses mildly differ. However, the method presented in this chapter cannot handle more challenging head rotations (for instance, pitch/yaw angles above 30 degrees) and can also fail if both the source and target actor move the head in completely different directions. Robust and accurate face tracking is addressed next in Chapter 5 and the advantages over purely image-based motion transfer are further illustrated in Chapter 6.

4.8 Summary

In this chapter, we have introduced an image-based reenactment system that replaces the inner face of an actor in a video, while preserving the original facial performance. The proposed method requires neither user interaction, nor a complex 3D face model. It is based on expression matching and uses temporal clustering for matching stability and a combined appearance and motion metric for matching coherence. A simple, yet effective, image-warping technique that allows us to deal with moderate head motion has also been presented. At the core of this method is an accurate localization of a sparse set of 2D facial landmarks based on optical flow between automatically selected keyframes. This enables us not only to compute representative appearance descriptors, but also to accurately detect and replace 2D faces in a video. Experiments show that convincing reenactment results for existing footage can be obtained by using only a short input video of a user making arbitrary facial expressions.

The results presented in this chapter have shown the first step towards face digitization in unconstrained videos, but they still lack the amount of detail and quality needed to create photo-realistic full 3D avatars. Improvements in the capture of digital models are presented next in Chapter 5.

Chapter 5

Model-based Face Capture in Semi-Constrained Setups

Figure 5.1: Two results obtained with the proposed method. *Left:* Input video. *Middle:* Tracked 3D mesh, overlaid over the input video. *Right:* By applying texture to the mesh and re-rendering it with the estimated scene lighting, a virtual face make-up effect can be produced.

Chapter 4 presented a robust image-based system for digital face reenactment in monocular videos. Compared to complex multiview-based systems (see Section 3.1.1), our reenactment method excels in simplicity of use, but it is still challenged by strong facial motion and head rotations, and also did not capture a full 3D model. This chapter pushes the boundaries of face digitization, especially capture, and presents a model-based approach for reconstructing detailed, spatio-temporally coherent 3D face geometry (see Figure 5.1) as well as scene lighting from 2D video footage. The proposed approach assumes that the camera's intrinsics are known and that a 3D reconstruction of the actor's face is available to create a personalized model. This renders high-quality monocular face capture more tractable. The method and results presented in this chapter are based on [Garrido et al. 2013].

5.1 Introduction

Optical performance capture methods can reconstruct faces of virtual actors in videos to deliver detailed dynamic face geometry. However, existing approaches are expensive and cumbersome as they may require dense multiview camera systems, controlled light setups, active markers in the scene, and recording in a controlled studio (Section 3.1.1). At the other end of the spectrum are computer vision methods that capture face models from monocular video (Section 3.1.3). These captured models are extremely coarse, and usually only contain sparse collections of 2D or 3D facial landmarks rather than a detailed 3D shape. Recently, Valgaerts et al. [2012b] presented an approach for detailed performance capture from binocular stereo. However, 3D face models of a quality level needed for movies and games cannot yet be captured from monocular video.

In an attempt to push the boundary and application range further, in this chapter we propose a new method to automatically capture *detailed* dynamic face geometry from *monocular* video filmed under general lighting. It fills an important algorithmic gap in the spectrum of performance capture techniques between expensive controlled setups and low-quality monocular approaches, and opens up new possibilities for professional movie and game productions by enabling performance capture on set, directly from the primary camera. Finally, it is a step towards democratizing face capture technology for everyday users with a single inexpensive video camera. Such is the relevance of high-quality face capture from monocular video that our method has inspired follow-up work in this direction (see Section 3.1.3).

A 3D face model for a monocular video is also a precondition for many relevant video editing tasks. Examples include face transfer [Vlasic et al. 2005], face replacement [Alexander et al. 2010], facial animation retiming [Dale et al. 2011] or puppeteering [Kemelmacher-Shlizerman et al. 2010; Li et al. 2012]. For the results obtained with these methods, a tracked geometry model of moderate shape detail was sufficient, but even then, substantial manual work is unavoidable to obtain a 3D face model that overlays sufficiently with the video footage. To achieve a higher quality of edits on more general scenes, and to show advanced edits such as relighting or virtual make-up, we require much higher detailed reconstructions from a single video.

Our approach relies on several algorithmic contributions that are joined with state-of-the-art 2D/3D vision and graphics techniques adapted to monocular video. In a preparatory step, we create a personalized blendshape model for the captured actor by transferring generic blendshapes to a static 3D face scan of the subject. This task is the only step requiring manual interaction. In the first step, we accurately track a sparse set of 2D facial features throughout the video using a state-of-the-art non-rigid feature tracking algorithm [Saragih et al. 2011a], but enhanced with a novel correction method presented in Section 4.3. After 2D landmark tracking, we obtain the model parameters (expression and pose) by solving a constrained quadratic programming problem. To further refine the alignment of the face model, a non-rigid, temporally-coherent geometry correction is performed using a novel multi-frame variational optical flow approach. Finally, a shape-from-shading refinement approach adapted to monocular video reconstructs fine-scale geometric detail after estimating the scene lighting and face albedo.

We emphasize the simplicity and robustness of the proposed lightweight and versatile performance capture method. Even though multiview methods achieve higher reconstruction quality, the proposed approach is one of the first of its kind to capture long sequences of expressive face motion for scenarios where none of these other methods are applicable. As an additional benefit, our tracker estimates blendshape parameters that can be used by animators (important feature also advocated in previous work [Weise et al. 2011]). We show qualitative and quantitative results on several expres-

(a) (b) (c) (d) (e)

Figure 5.2: Algorithm overview: *Left to right:* (a) Input video frame, (b) 2D feature tracking **(Section 5.4.1)**, (c) expression and pose estimation using a blendshape model **(Section 5.4.2)**, (d) dense expression and pose correction **(Section 5.5)**, (e) shape refinement **(Section 5.6)**.

sive face sequences captured under uncontrolled lighting, both indoors and outdoors. Our approach compares favorably to the recent state-of-the-art binocular method of Valgaerts et al. [2012b], and even performs better for certain aspects. As a proof-of-concept example of advanced video editing, we show the application of virtual face texture to video (see Figure 5.1).

5.2 Overview

Our method uses as input a single video of a face, captured under unknown lighting with a camera that has precomputed intrinsics. It is composed of four main computational steps:

S0 **Personalized blendshape model creation (Section 5.3):** We construct a customized parametric 3D blendshape model for every actor, which is used to reconstruct all sequences starring that actor.

S1 **Blendshape tracking (Section 5.4):** We accurately track a sparse set of temporally stable 2D facial features throughout the monocular video using the method described in Section 4.3, see Figure 5.2 (b). From the established sparse feature set, we estimate a global 3D transformation (head pose) and a set of model parameters (facial expression) for the blendshape model, see Figure 5.2 (c).

S2 **Dense tracking correction (Section 5.5):** Next, we improve the facial expression and head pose obtained from the sparse blendshape tracking step by computing a temporally coherent and dense motion field in video. This motion field is then employed to correct the facial geometry to obtain a more accurate model-to-video alignment, see Figure 5.2 (d).

S3 **Dynamic shape refinement (Section 5.6):** In a final step, we reconstruct fine-scale, time-varying facial detail, such as wrinkles and folds. We do this by estimating the unknown lighting and exploiting shading cues for shape refinement, see Figure 5.2 (e).

Notation A frame in the monocular video corresponding to timestamp t will be denoted by f^t, with f^{t_0} being the starting frame. We reconstruct a spatio-temporally coherent sequence of triangular face meshes M^t, consisting of a fixed set of n vertices with Euclidean coordinates $\mathbf{X}^t$ and their connecting edges. The outcome of the subsequent computational steps in our algorithm are the

tracked mesh M_{b}^t (S1), the *corrected mesh* M_{c}^t (S2) and the final *refined mesh* M_{r}^t (S3), all sharing the same topology (i. e., vertex set and connectivity).

5.3 Personalized Blendshape Model Creation

We use a *delta blendshape model* as a parametric morphable 3D representation of the face (see Section 2.1.1 for further details). For the sake of simplicity, we will refer to this model as *blendshape model*. Let $\mathbf{b}_0 \in \mathbb{R}^{3n}$ be the neutral shape containing the coordinates of the n vertices of a face mesh at rest pose. A new facial expression $\mathbf{e}$ can be obtained by the linear combination:

$$\mathbf{e}(\beta_1, \ldots, \beta_k) = \mathbf{b}_0 + \sum_{j=1}^{k} \beta_j \mathbf{d}_j \ , \tag{5.1}$$

where $\mathbf{d}_j \in \mathbb{R}^{3n}$, with $1 \leq j \leq k$, are the blendshape displacements (i. e., delta blendshapes) and $0 \leq \beta_j \leq 1, \forall j$ are the k *blendshape weights*.

We create an actor specific face model by taking a generic, artist-created, professional blendshape model ($k = 78$) obtained from Faceware Technologies[1] and then performing a non-rigid registration of the neutral shape to a binocular stereo reconstruction [Valgaerts et al. 2012a] of the actor's face at rest pose. Please note that any generic blendshape model preferred by an artist and any laser scanning or image-based face reconstruction method[2] can be used instead. Registration is based on manually matching 29 3D landmarks on the eyes, nose and mouth, followed by a global correspondence search and Laplacian regularized shape deformation [Sorkine 2005]. Once the neutral shape is registered, the blendshapes of the generic model are transferred using the same procedure. The obtained face models have a person specific shape, but the same semantic dimensions are shared over all actors. Although our straightforward registration approach has proven sufficient for our application, additional person-specific semantics can be included by using extra scans of different expressions [Li et al. 2010]. Since all personalized blendshape models are derived from the same generic model, they share the same number of vertices (200k) and triangulation (henceforth shared by all meshes shown in this chapter). Figure 5.3 shows a selection of expressions for the generic Emily model and for the four corresponding personalized models derived from it. These four personalized models were used to generate the results shown in Figure 5.7, Figure 5.8, Figure 5.9, and Figure 5.10. Note that the produced blendshape models lack in high frequency shape detail, such as wrinkles and folds.

5.4 Blendshape Tracking

5.4.1 Accurate 2D Facial Feature Tracking

An essential part in the tracking is the detection of a sparse set of m accurate, temporally stable 2D facial features (i. e., landmarks) that serves as the base of our approach to find an initial coarse alignment of the personalized 3D blendshape model to an image at frame f^t. In our approach, we accurately track $m = 66$ landmarks on the actor's face using the method described in Section 4.3.

[1] www.facewaretech.com
[2] www.facegen.com

Figure 5.3: Personalized blendshape models. *Top row:* 9 out of 79 expressions of a blendshape model created by an artist, including the neutral pose (courtesy of Faceware Technologies). The neutral expression is shown on the left. *Next four rows:* The same 9 expressions transferred to the personalized models of the actors used in our four test sequences (see Figure 5.7, Figure 5.8, Figure 5.9, and Figure 5.10). All meshes share the same topology, i. e., number of vertices, triangulation, and connectivity. Besides, all models span the same semantic dimensions.

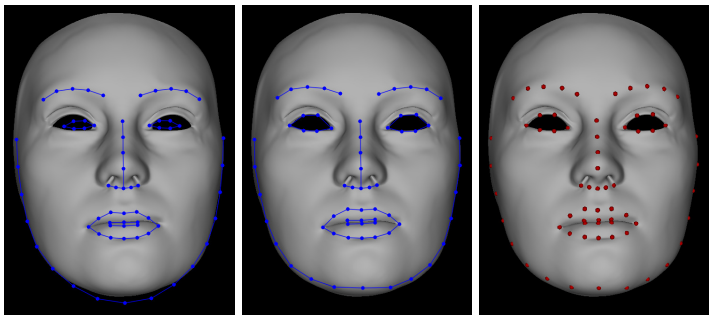


Figure 5.4: Coupling the 2D and 3D model. *Left:* Features estimated by an off-the-shelf feature tracker (see Section 4.3) for a rendered frontal view of the generic blendshape at rest pose. *Middle:* The manually corrected features. *Right:* The 3D feature vertices on the generic blendshape model.

5.4.2 Coarse Expression and Rigid Pose Estimation

We now align the 3D blendshape model to the sparse set of 2D feature locations found in each frame: We solve an optimization problem in the blendshape space to find the pose and facial expression parameters of the 3D face model, such that it optimally reprojects onto the tracked 2D feature locations. This is performed in three steps, described as follows.

Coupling the 2D and 3D Model

To couple the m 2D feature points that are tracked in the video to their corresponding 3D positions on the generic blendshape model, we render a frontal snapshot of the neutral pose in OpenGL using a standard phong reflection model with a single point light source pointing towards the face surface. Once rendered, we run an off-the-shelf feature tracking algorithm to estimate the position of the main facial features[3] [Saragih et al. 2011a]. The detection works relatively well for a shaded rendering of the model with constant material in front of a black background, but the detected features still need minor manual correction for better alignment (see Figure 5.4). For the Emily blendshape model, the eyes are unnaturally large and particularly these detected features need further correction. As the 2D features are the projections of the corresponding 3D points on the blendshape model, correspondences can be easily established by back projection on the mesh. From now on, we will denote these 3D feature points as F.

Since all personalized blendshape models are derived from the same generic Emily model, the indices of the found set of 3D feature vertices remain the same for all actors. Thus, this step needs to be completed just once and has to be repeated only if a different generic face model is used.

Expression Estimation

Given a set of 2D facial feature locations $\mathbf{x}_i^t$, $1 \leq i \leq m$ estimated in the current frame f^t, and a personalized blendshape model $\mathbf{e}(\beta_1, \ldots, \beta_k)$, our task is to estimate the current facial expression in terms of the blendshape weights β_j^t, $1 \leq j \leq k$. This expression transfer problem can be formulated

[3] Note that this is the same tracking algorithm mentioned in Section 4.3 for which we improve the landmark trajectories.

in a constrained least squares sense, as follows:

$$\min_{\beta_j^t} \sum_{i=1}^{m} \left\| \left(s^t R^t P^\top \mathbf{x}_i^t + \mathbf{t}^t \right) - \mathbf{X}_{F,i}\left(\beta_j^t\right) \right\|_2^2 \quad \text{with} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \ , \tag{5.2}$$

$$\text{s.t.:} \quad 0 \leq \beta_j^t \leq 1 \quad \text{for} \quad 1 \leq j \leq k \ , \tag{5.3}$$

where $\mathbf{X}_{F,i} \in F$ are the coordinates of the feature vertices of the blendshape model, P is the orthogonal weak perspective projection matrix. s^t, $R^t \in \mathbb{R}^{3\times3}$ and $\mathbf{t}^t \in \mathbb{R}^3$ denote the global scaling factor, the rotation matrix, and the translation vector which align the reprojected feature locations with the feature vertices of the blendshape model in a weak perspective camera model setting (see Section 2.2.1 for further details). Since the alignment transformations are unknown, we solve the above quadratic programming problem iteratively: First we optimize for $\{s, R, \mathbf{t}\}^t$ using a current estimate for the blendshape weights, after which we solve for β_j^t in a second step keeping the transformations fixed. We terminate when the change in β_j^t falls below a threshold.

Solving for the Transformations Finding the least squares solution of $\{s, R, \mathbf{t}\}^t$ to Equation 5.2 for a constant set of blendshape weights is equivalent to aligning two 3D point sets, which can be solved in closed form by singular value decomposition (SVD) [Arun et al. 1987].

Solving for the Blendshape Weights Once the alignment transformations have been computed, we search for an optimal combination of the linear weights β_j^t that minimizes the difference in shape between the point sets $\left(s^t R^t P^\top \mathbf{x}_i^t + \mathbf{t}^t\right)$ and $\mathbf{X}_{F,i}\left(\beta_j^t\right)$, $1 \leq i \leq m$, subject to the box constraints shown in Equation 5.3. By rewriting the blendshape model (see Equation 5.1) as:

$$\mathbf{e}(\beta_j) = \left(1 - \sum_{j=1}^{k} \beta_j\right) \mathbf{b}_0 + \sum_{j=1}^{k} \beta_j \left(\mathbf{b}_0 + \mathbf{d}_j\right) \ , \tag{5.4}$$

and defining $\beta_0 = 1 - \sum_{i=j}^{k} \beta_j$, we obtain an instance of a convex quadratic programming problem with box constraints and a linear equality constraint. This can be solved efficiently by methods based on sequential minimal optimization[4] [Platt 1998]. Note that the solver we used implicitly enforces L_1 regularization on the estimated weights.

As opposed to the alignment step, we found experimentally that the blendshape weight optimization is more robust if it is only performed over the X- and Y-coordinates. As such, we discard depth information in this step.

3D Pose Estimation

To retrieve the head pose under a full perspective projection, we update the positions of the 3D feature vertices in F using the computed blendshape weights, and feed them together with the tracked 2D facial feature locations to a pose estimation algorithm [David et al. 2004]. This algorithm approximates the perspective projection by a series of scaled orthographic projections and iteratively estimates the global pose parameters for the given set of 2D-to-3D correspondences. Note that this algorithm assumes that the camera's intrinsics, i. e., focal length f and principal point $\mathbf{c}$ are known beforehand. For each sequence presented in Section 5.7.1, these parameters were estimated in a

[4] http://cmp.felk.cvut.cz/~xfrancv/libqp/html/

Figure 5.5: Dense expression and pose correction. *Left:* Overlay of the tracked blendshape model of Figure 5.2 (c), textured with the starting frame. *Middle:* Textured overlay of the tracking-corrected face mesh of Figure 5.2 (d). This synthetic frame is closer to the target frame in Figure 5.2 (a). *Right:* Per-vertex correction represented as a heatmap on the corrected mesh, where red means large correction and green means small correction.

pre-calibration step using the MATLAB calibration toolbox, which requires the user to rotate and move a checkerboard pattern in front of the camera for a few seconds.

Expression and pose estimation are iterated until convergence, resulting in a *tracked face mesh* M_{b}^{t} with associated blendshape weights and pose parameters. However, M_{b}^{t} lies within the space spanned by the blendshape model and lacks high-frequency face detail that appears in the video. These shortcomings will be tackled next.

5.5 Dense Tracking Correction

After coarse expression and pose estimation, there may remain residual errors in the facial expression and head pose which can lead to misalignments when overlaying the 3D model with the video, see Figure 5.5. The first reason for this error is that the used parametric blendshape model has a limit in expressibility and is not able to exactly reproduce a target expression that is not spanned by its basis of variation. The second reason is that the optimization of the previous section is performed over a sparse, fixed set of feature vertices and excludes vertices that lie in other facial regions, such as the cheeks or the forehead. To obtain an accurately aligned 3D mesh, we correct the initially estimated expression and pose over all vertices.

5.5.1 Temporally Coherent Corrective Flow

To correct the expression and pose of the face mesh M_{b}^{t}, obtained by blendshape tracking, we assign a *fixed* color to each vertex using projective texturing and blending from the starting frame f^{t_0}. Projecting M_{b}^{t} back onto the image plane at every time t results in the synthetic image sequence f_{s}, depicted in Figure 5.6. To ensure optimal texturing for the results presented in Section 5.7.1, we manually improved the detected feature locations in the starting frame.

The idea behind our correction step is to compute the dense optical flow field that minimizes the difference between a synthetic frame f_{s}^{t} and its corresponding true target frame f^{t}, and then use the flow to deform the mesh. This corrective optical flow is denoted as $\mathbf{w}_1$ in Figure 5.6. Computing such corrective optical flow independently for each time t introduces temporal artifacts in the corrected mesh geometry due to the lack of coherence over time in the optical flow estimation. An

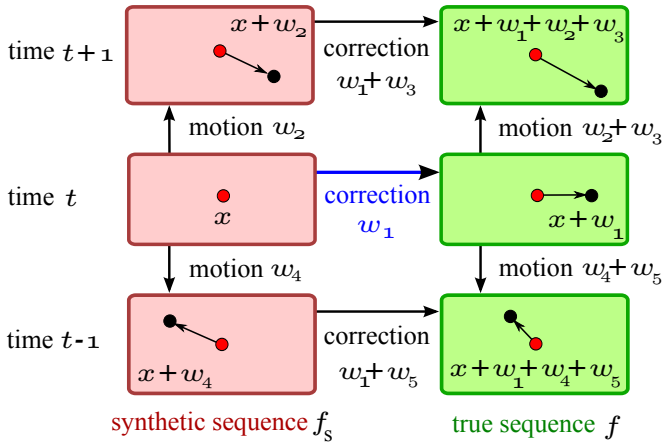


Figure 5.6: Temporally-coherent corrective flow estimation.

illustration of such temporal artifacts can be seen in the second supplementary video at the project website[5]. Let us now assume that M_b^t deforms coherently and smoothly over time, and so does the synthetic sequence. Consequently, the corrective flow $\mathbf{w}_1$ between f_s^t and f^t has to vary gradually over time as well, since the true sequence is smooth by construction.

To impose temporal smoothness on $\mathbf{w}_1$, we include frames at $t+1$ and $t-1$ and introduce a new optical flow method for the six-frame scenario depicted in Figure 5.6. By exploiting the dependencies between the correspondences, the problem can be parametrized w. r. t. the reference frame f_s^t by $\mathbf{w}_1$ and four additional flows: $\mathbf{w}_2$ and $\mathbf{w}_4$ describing the face motion in the synthetic sequence, and $\mathbf{w}_3$ and $\mathbf{w}_5$ describing the temporal change in the corrective flow $\mathbf{w}_1$. Note that $\mathbf{w}_1+\mathbf{w}_3$ and $\mathbf{w}_1+\mathbf{w}_5$ represent the corrective flows in the corresponding image points at $t+1$ and $t-1$. Thus, we can impose temporal coherence through the flow changes $\mathbf{w}_3$ and $\mathbf{w}_5$.

To estimate all unknown flows simultaneously, we minimize an energy consisting of data, smoothness, and similarity constraints, as follows:

$$E = \int_\Phi \left(\sum_{i=1}^{7} E_{\text{data}}^i + \sum_{i=1}^{5} \alpha_i E_{\text{smooth}}^i + \sum_{i=1}^{2} \gamma_i E_{\text{sim}}^i \right) \mathrm{d}\mathbf{x} \ , \tag{5.5}$$

where $\Phi \in \mathbb{R}^2 = \{x, y \in \mathbb{R} | x, y \geq 0\}$ represents the rectangular image domain, and α_i, γ_i, $\forall i$ are trade-off factors that control the amount of smoothness and similarity, respectively.

Data Constraints The data terms of the energy shown in Equation 5.5 impose photometric constancy between corresponding points along the seven connections drawn in Figure 5.6. For bright-

[5] http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/

ness constancy, the data terms take the following form:

$$E^1_{\text{data}} = \Psi_{\text{d}}\left(|f^{t+1}(\mathbf{x}+\mathbf{w}_1+\mathbf{w}_2+\mathbf{w}_3) - f^{t+1}_{\text{s}}(\mathbf{x}+\mathbf{w}_2)|^2\right) \ , \tag{5.6}$$

$$E^2_{\text{data}} = \Psi_{\text{d}}\left(|f^{t}(\mathbf{x}+\mathbf{w}_1) - f^{t}_{\text{s}}(\mathbf{x})|^2\right) \ , \tag{5.7}$$

$$E^3_{\text{data}} = \Psi_{\text{d}}\left(|f^{t-1}(\mathbf{x}+\mathbf{w}_1+\mathbf{w}_4+\mathbf{w}_5) - f^{t-1}_{\text{s}}(\mathbf{x}+\mathbf{w}_4)|^2\right) \ , \tag{5.8}$$

$$E^4_{\text{data}} = \Psi_{\text{d}}\left(|f^{t+1}_{\text{s}}(\mathbf{x}+\mathbf{w}_2) - f^{t}_{\text{s}}(\mathbf{x})|^2\right) \ , \tag{5.9}$$

$$E^5_{\text{data}} = \Psi_{\text{d}}\left(|f^{t-1}_{\text{s}}(\mathbf{x}+\mathbf{w}_4) - f^{t}_{\text{s}}(\mathbf{x})|^2\right) \ , \tag{5.10}$$

$$E^6_{\text{data}} = \Psi_{\text{d}}\left(|f^{t+1}(\mathbf{x}+\mathbf{w}_1+\mathbf{w}_2+\mathbf{w}_3) - f^{t}(\mathbf{x}+\mathbf{w}_1)|^2\right) \ , \tag{5.11}$$

$$E^7_{\text{data}} = \Psi_{\text{d}}\left(|f^{t-1}(\mathbf{x}+\mathbf{w}_1+\mathbf{w}_4+\mathbf{w}_5) - f^{t}(\mathbf{x}+\mathbf{w}_1)|^2\right) \ , \tag{5.12}$$

where $f(\mathbf{x}) = [f(\mathbf{x})^r, f(\mathbf{x})^g, f(\mathbf{x})^b]^\top$ is the pixel color at position $\mathbf{x} = [x,y]^\top$ in the image and $\Psi_{\text{d}}(s^2) = \sqrt{s^2 + (0.001)^2}$ is the robust regularized L_1 penalizer. To make the flow estimation more robust to gradual lighting changes in the scene and improve the overall matching accuracy, all constraints are extended with a gradient constancy assumption and color information. For instance, the second data term between f^t and f^t_{s}, and fourth data term between f^{t+1}_{s} and f^t_{s} can be written as follows:

$$E^2_{\text{data}} = \Psi_{\text{d}}\left(|f^{t}(\mathbf{x}+\mathbf{w}_1) - f^{t}_{\text{s}}(\mathbf{x})|^2 + \zeta|\nabla f^{t}(\mathbf{x}+\mathbf{w}_1) - \nabla f^{t}_{\text{s}}(\mathbf{x})|^2\right) \ , \tag{5.13}$$

$$E^4_{\text{data}} = \Psi_{\text{d}}\left(|f^{t+1}_{\text{s}}(\mathbf{x}+\mathbf{w}_2) - f^{t}_{\text{s}}(\mathbf{x})|^2 + \zeta|\nabla f^{t+1}_{\text{s}}(\mathbf{x}+\mathbf{w}_2) - \nabla f^{t}_{\text{s}}(\mathbf{x})|^2\right) \ , \tag{5.14}$$

where $\zeta \geq 0$ is a weighting factor and $\nabla = [\partial_x, \partial_y]^\top$ denotes the spatial gradient operator. We empirically found that $\zeta = 0.1$ yields a good compromise [Valgaerts et al. 2010].

Smoothness Constraints Similar in spirit to the scene flow scenario presented by Valgaerts et al. [2012b], we use a structure-aware regularization for the flows $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_4$ to improve the optical flow estimation in semantically meaningful regions of the face, namely:

$$E^1_{\text{smooth}} = \Psi_{\text{s}}\left(|\nabla \mathbf{w}_1^\top \mathbf{r}_1|^2\right) + \Psi_{\text{s}}\left(|\nabla \mathbf{w}_1^\top \mathbf{r}_2|^2\right) \ , \tag{5.15}$$

$$E^2_{\text{smooth}} = \Psi_{\text{s}}\left(|\nabla \mathbf{w}_2^\top \mathbf{r}_1|^2\right) + \Psi_{\text{s}}\left(|\nabla \mathbf{w}_2^\top \mathbf{r}_2|^2\right) \ , \tag{5.16}$$

$$E^4_{\text{smooth}} = \Psi_{\text{s}}\left(|\nabla \mathbf{w}_4^\top \mathbf{r}_1|^2\right) + \Psi_{\text{s}}\left(|\nabla \mathbf{w}_4^\top \mathbf{r}_2|^2\right) \ , \tag{5.17}$$

where the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ denote orthogonal smoothing directions along and across flow structures, and $\Psi_{\text{s}}(s^2) = 2\lambda_{\text{s}}^2\sqrt{1+(s/\lambda_{\text{s}})^2}$, $\lambda_{\text{s}} = 0.1$ is a discontinuity-preserving function. As opposed to the corrective and motion flows, we regularize $\mathbf{w}_3$ and $\mathbf{w}_5$ much stronger using L_2 regularization:

$$E^3_{\text{smooth}} = |\nabla \mathbf{w}_3|^2 \quad \text{and} \quad E^5_{\text{smooth}} = |\nabla \mathbf{w}_5|^2 \ . \tag{5.18}$$

This quadratic regularization of the flow changes ensures that the corrective flow $\mathbf{w}_1$ varies smoothly over time.

Similarity Constraints Finally, we enforce the corrective flows $\mathbf{w}_1$, $\mathbf{w}_1+\mathbf{w}_3$ and $\mathbf{w}_1+\mathbf{w}_5$ to be similar to each other, i. e., we strongly penalize the magnitude of the flow changes:

$$E^1_{\text{sim}} = |\mathbf{w}_3|^2 \quad \text{and} \quad E^2_{\text{sim}} = |\mathbf{w}_5|^2 \ . \tag{5.19}$$

The terms in Equation 5.19 and Equation 5.18 can be related to first and second order smoothness constraints along optical flow trajectories, as described in [Volz et al. 2011]. Contrary to their approach, we exploit the circular dependencies in our specific set-up for the purpose of coherently correcting one image sequence w. r. t. another.

The total energy shown in Equation 5.5 is minimized over all flows by a coarse-to-fine multiresolution strategy using a non-linear multigrid method [Papenberg et al. 2006]. Computational time can be sped up by utilizing the forward and backward optical flows computed in the non-rigid tracking step (see Section 4.3) as initialization.

5.5.2 Optical Flow-based Mesh Deformation

We correct the geometry of M_{b}^t by projecting the estimated optical flow $\mathbf{w}_1$ back onto the mesh and retrieving a corrective 3D motion vector for each vertex. Since our monocular setting has an inherent depth ambiguity, it is impossible to recover the correct motion in the Z-direction (i. e., in depth). However, we experienced that correcting each vertex in X- and Y-directions parallel to the image plane still produces realistic and expressive results. In Chapter 7, we overcome this limitation by proposing a parametric corrective field based on harmonics functions that parametrize true 3D displacements. Such dense parametric 3D correction will be then learned as a function of the blendshape weights to infer person-specific expressions for a detailed, personalized blendshape rig (see Chapter 8).

Let us denote $\mathbf{W}^t \in \mathbb{R}^{n\times 3}$ as the 3D motion field parallel to the image plane. We use $\mathbf{W}^t$ to propagate each vertex to its new position in the *corrected face mesh* M_{c}^t. To ensure a smooth deformation, we minimize the following Laplacian-regularized energy:

$$E = \left\| LX_{\mathrm{c}}^t - LX_{\mathrm{b}}^t \right\|^2 + \mu^2 \sum_{i\in C^t} \left\| \mathbf{X}_{\mathrm{c},i}^t - (\mathbf{X}_{\mathrm{b},i}^t + \mathbf{W}_i^t) \right\|^2 \quad , \tag{5.20}$$

where $L \in \mathbb{R}^{n\times n}$ is the Laplacian matrix of M_{b}^t computed with cotangent weights [Sorkine 2005], X_{c}^t and $X_{\mathrm{b}}^t \in \mathbb{R}^{n\times 3}$ represent the matrices collecting the positions of all vertices $\mathbf{X}_i^t$ in M_{c}^t and M_{b}^t, $1 \leq i \leq n$, and μ is a trade-off weight. The set C^t is a uniformly subsampled selection of visible vertices at frame f^t.

We perform the steps of Section 5.5.1 and Section 5.5.2 once per frame, but they could be applied iteratively. Note that this correction takes us slightly outside the 3D shape space spanned by the blendshape model and yield an extremely accurate alignment of the mesh with the video. The alignment before and after correction is shown in Figure 5.5.

5.6 Dynamic Shape Refinement

In a final step, we capture and add fine-scale surface detail to the tracked mesh, such as emerging or disappearing wrinkles and folds. Our approach is based on the *shape-from-shading* framework under general unknown illumination that was proposed in [Valgaerts et al. 2012b] for the binocular reconstruction case. At a given frame f^t, the method first estimates the unknown incident lighting based on an estimate of geometry and coarse albedo. The estimated lighting is in turn used to deform the geometry, such that the rendered shading gradients and the image shading gradients agree. Essentially, this method inverts the rendering equation to be able to reconstruct the scene,

which is much easier in a setting with multiple cameras, since the face surface is seen from several viewpoints, and therefore it constrains the solution space better.

To adjust this approach to the monocular case, we estimate the unknown illumination from a larger temporal baseline to compensate for the lack of additional cameras. In our setting, we assume that the illumination conditions do not change over time. However, a ground truth light probe to simulate the static light environment is not available and must be estimated. To tackle this problem, we first estimate lighting, albedo and refined surface geometry of the tracked face mesh for the first 10 frames of every video using the exact same approach as [Valgaerts et al. 2012b]. In our monocular case, since the estimation is much more under-constrained and error-prone, we only use this result as an initialization. In a second step, we jointly use the initial albedo and fine scale geometry to estimate a single environment map that globally fits to all timesteps in the small subsequence. We then use this static light environment and estimate the dynamic geometry detail at each timestep [Valgaerts et al. 2012b]. The result of dynamic shape refinement is the final *refined face mesh* M_{r}^{t}.

To further remove temporal flicker in the visualization of the results, we update the surface normals by averaging them over a temporal window of size 5 and adapt the geometry to the updated normals, as proposed in [Nehab et al. 2005].

It is important to remark that in Chapter 8 fine-scale surface detail will be learned as a function of the blendshape weights to infer a personalized fine-scale skin detail layer that dynamically correlates to facial expressions, thus giving an extra layer of personalization to the reconstructed blendshape rig.

5.7 Experiments

We evaluated the performance of our approach on four video sequences of different actors with lengths ranging from 565 (22 s) to 1000 frames (40 s). Three videos were recorded indoors with a Canon EOS 550D camera at 25 fps in HD quality (1920×1088 pixels) and one video was recorded outdoors with a GoPro camera at 30 fps in HD quality. Our approach was implemented in C++ and tested on a 3.4 GHz Intel®Core™ i5 processor with 16GB RAM. All the results shown below are viewed best as video. Hence, the reader is strongly encouraged to watch the supplemental videos at the project website[6].

For all results, λ_i was 0.1 for the mouth features, 0.5 for the eye features, and 0.2 for the remaining features. For the Canon results, $\alpha_1 = 500$, $\alpha_2 = \alpha_4 = 600$, and $\alpha_3 = \alpha_5 = 300$, and for the GoPro result $\alpha_2 = \alpha_4 = 700$, and $\alpha_3 = \alpha_5 = 400$. Furthermore, $\gamma_1 = \gamma_2 = 50$ and $\mu = 0.5$. For improved accuracy around the eye lids, the eyes of the blendshape model were filled before tracking, but not visualized in the final results. Eye filling is only done once in the generic model and does not change any step of our method.

5.7.1 Results

Performance Capture The first two results are part of a calibrated binocular stereo sequence recorded under uncontrolled indoor lighting [Valgaerts et al. 2012b]. We only use one camera output for our method and need one extra frame from the second camera for the blendshape model creation. Results for the first sequence, featuring very expressive gestures and normal speech, are

[6] http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/

Figure 5.7: Results for expressive facial motions - first sequence, 565 frames (22 s). *Left to right:* The input frame, the corresponding blended overlay of the reconstructed mesh, a 3D view of the mesh, and an example of applying virtual face texture using the estimated geometry and lighting.

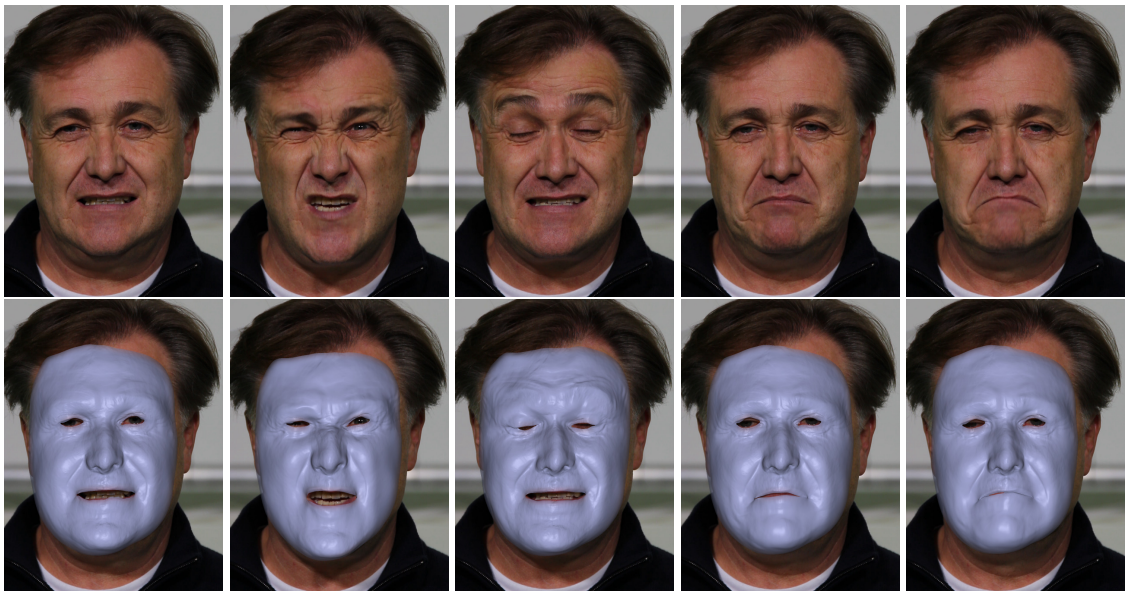


Figure 5.8: Results captured indoors - second sequence, 620 frames (25 s). This sequence was recorded with a Canon EOS 550D camera and exhibits expressive and fast facial gestures.

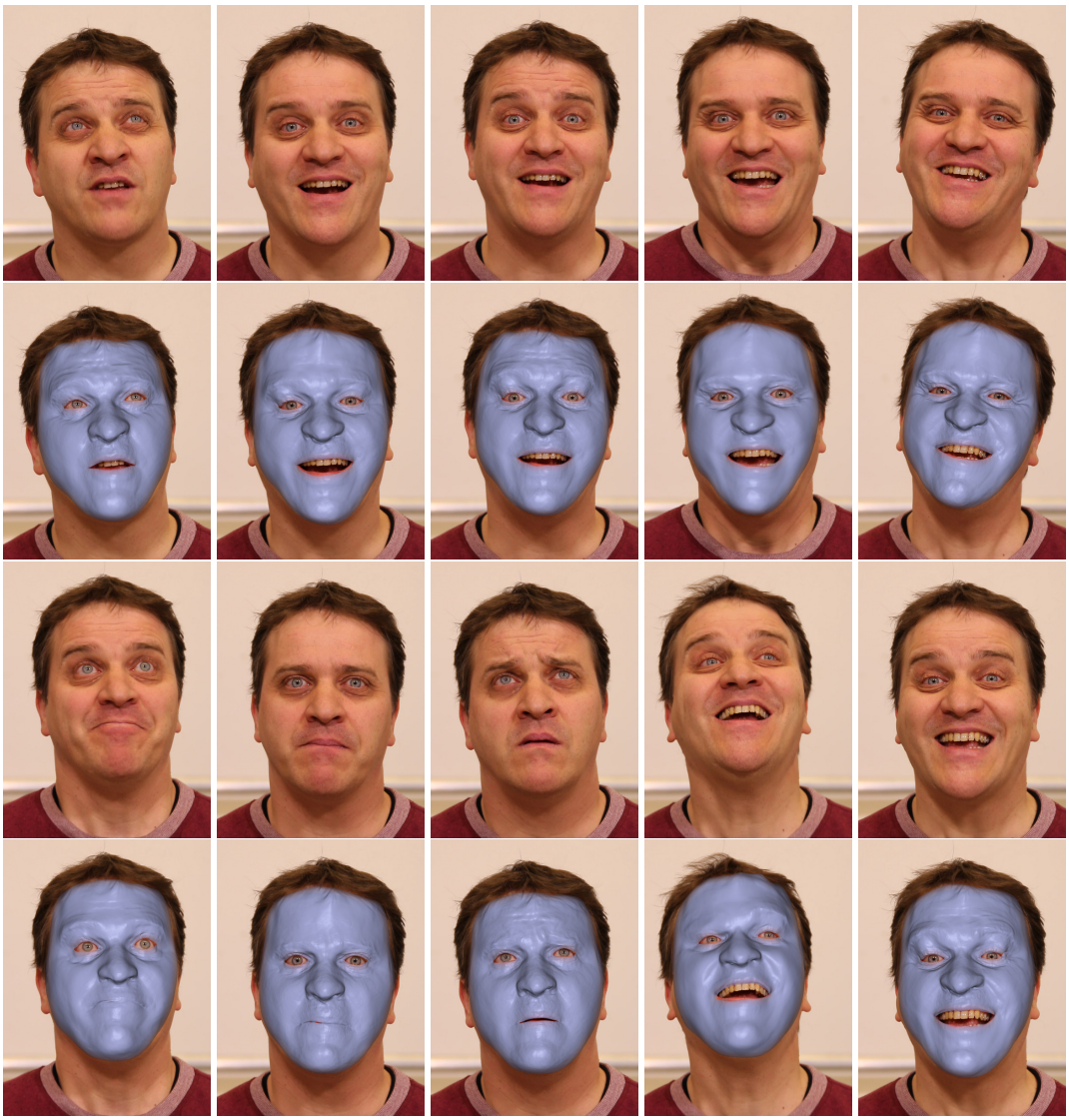


Figure 5.9: Results captured indoors - third sequence, 1000 frames (40 s). This sequence was recorded with a Canon EOS 550D camera and shows expressive faces, fast facial motion, and challenging head movement.

shown in Figure 5.7. All meshes consist of the same set of vertices and are produced by tracking and refining the personalized blendshape model of Figure 5.3 (second row) over 565 frames. The green screen is part of the recording and is not used. The figure shows that we are able to faithfully capture very challenging facial expressions, even for gestures that are not spanned by the blendshape model, e. g., the bottom row. The third column illustrates that our method effectively reconstructs a space-time coherent 3D face geometry with dynamic fine scale detail. Although the actor's head hardly moves in depth, our method estimates a small global translation component in the camera direction, which we discard for the 3D visualization in the figures. Figure 5.8 shows a result for a second sequence of 620 frames, featuring fast and expressive motion. Our results capture a high level of shape, motion, and surface detail.

Figure 5.9 shows an additional result for a third sequence, newly recorded under similar conditions as the first two. The sequence depicts a recitation of a theatrical play and is extremely challenging

Figure 5.10: Results captured with a hand-held GoPro camera - fourth sequence, 650 frames (22 s). This sequence was recorded outdoors under unknown lighting and features challenging head motion. The rightmost column shows a failure case where our method does not estimate the pose and expression correctly. The supplementary video at the project website shows that our method fully recovers afterwards.

due to its length of 1000 frames, its diversity in facial expressions, and its fast and shaky head motion. The overlays in the figure show that we are able to estimate the X- and Y-components of the head pose very accurately and retrieve very subtle facial expressions, demonstrating the applicability of our method for demanding real world applications. Finally, we also captured an actor's facial performance outdoors with a lightweight GoPro camera. Despite the low quality of the video and the uncontrolled setting, we obtain accurate tracking results and realistic face detail, see Figure 5.10. This figure also shows a limitation of our approach for extreme out-of-plane head rotations, e. g., extreme pitch. However, the supplemental video available at the project website demonstrates how the algorithm fully recovers once the head pose comes back to a less extreme pose.

Virtual Face Texture As our capturing process introduces very little perceivable drift (see checkerboard texture in the supplemental video at the project website), it is well suited for video augmentation tasks such as adding virtual textures or tattoos[7], as shown in Figure 5.1 and 5.7. To this end, we render the texture as a diffuse albedo map on the moving face and light it with the estimated incident illumination. The texture is rendered in a separate channel and overlaid with the input video using Adobe Premiere. Our detailed reconstruction and lighting of the deformation detail is important to make the shading of the texture correspond to the shading in the video, giving the impression of virtual make-up.

Runtimes For the Canon sequences, the blendshape tracking and tracking correction run at a respective speed of 10 s and 4 min per frame, whereas the shading-based refinement has a run time of around 5 min per frame. All three steps run fully automatically and can be started in parallel with a small frame delay. The only tasks that require user intervention are the creation of the personalized blendshape model (Section 5.3, about 20 min), the one-time 2D-to-3D model coupling (Section 5.4.2, around 10 min) and the texturing of the blendshape model (Section 5.5.1,

[7]Design taken from www.deviantart.com/ under a CC license.

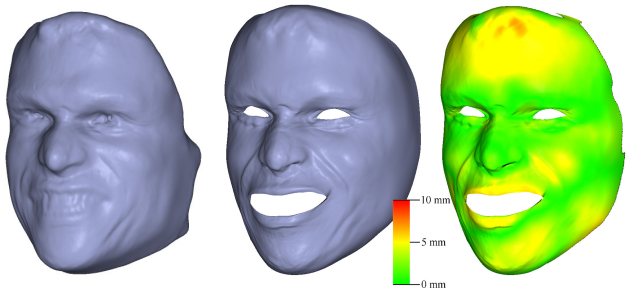


Figure 5.11: Comparison with the binocular method of Valgaerts et al. [2012b]. *From left to right:* Binocular reconstruction for the frame shown in Figure 5.2. Our reconstruction. Color-coded error between both reconstructions, represented as the per-vertex Euclidean distance (see error scale).

Table 5.1: Quantitative comparison. Average Euclidean distance between the binocular and monocular reconstructions computed on the first sequence (see Figure 5.7) and the second sequence (see Figure 5.8). The distance was computed between the nearest vertices of both meshes, but only over visible regions. This Euclidean distance is visualized in the figures as a heatmap overlay.

Sequence	Average distance (mm)	Average maximum distance (mm)
First sequence (over 565 frames)	1.71	7.45
Second sequence (over 402 frames)	2.91	9.82

around 10 min).

5.7.2 Validation

Comparison with Binocular Reconstruction In Figure 5.11, we quantitatively compare our results with a binocular facial performance capture method [Valgaerts et al. 2012b]. In the middle and right panes, we show our reconstructed face mesh for the target frame of Figure 5.2 and its deviation w. r. t. the corresponding binocular result shown on the left. Note that the color-coded error plots shown in the figure depict the Euclidean distance between the nearest visible vertices on the binocular and monocular meshes, and were produced by first aligning the meshes at their initial frames using rigid ICP and then tracking them throughout the sequence, while discarding the small translation in the depth direction. As the figure illustrates, errors mainly appear near the lips, cheeks and forehead, stemming from depth inaccuracies that the dense expression correction approach presented in Section 5.5 cannot refine.

Table 5.1 reports average errors for two indoor sequences. The geometric error (i. e., per-vertex Euclidean distance between two meshes) was computed as described above. Note that the deviation of our monocular result from the binocular results lies in the millimeter range despite the lack of direct depth information.

Another qualitative comparison between our monocular method and the binocular approach is shown in Figure 5.12. Here, this particular frame depicts fast rotating head motion. As reported by Valgaerts et al. [2012b], purely mesh-based binocular methods are sensitive to occlusions and drift in the presence of strong apparent out-of-plane head rotation, leading to unnatural deformations in some frames. Our monocular method, on the other hand, robustly tracks a parametric face model

Figure 5.12: Comparison of our method with the binocular method of Valgaerts et al. [2012b] for the results computed on the third sequence. *Left to right:* Target frame showing fast head rotation, result obtained by the binocular approach, and our result.

and only leaves the blendshape space in the expression correction step by computing a small deformation field. Hence, our model-based method is less susceptible to occlusions and drift, and overall, it is more robust to extreme head motions.

5.8 Discussion and Limitations

Our face tracking and refinement method is automatic, but creating the personalized blendshape model and improving the 2D features in the first frame for texturing rely on a small amount of user interaction. This is because each of these tasks corresponds to a hard computer vision sub-problem. Currently, our optical flow-based correction only uses textures at neutral pose to avoid including transient high-frequency details, although a non-rest texture could be used as well (albeit a bit harder engineering task). Furthermore, we assume that a stereo 3D reconstruction of the actor is available to create a personalized blendshape model, and take for granted that the camera's intrinsics can be calibrated in a pre-processing step. Most of the challenges presented above are approached in Chapter 7.

The proposed method attains very detailed and expressive results, but it is not completely free of artifacts. The dynamic texture example shown in the supplementary video at the project website[8] illustrates that small tracking inaccuracies can still be observed, e. g., around the teeth and lips. Small tangential floating of the vertices may also be present, as observed in the virtual texture overlays and the dynamic texture in the UV domain. For the GoPro result, artifacts around the nose are visible due to the challenging low-quality input (noise, rolling shutter, and color saturation). Extremely fast motion can be problematic for feature tracking with optical flow and our method currently does not handle light changes as it violates the optical flow assumptions. Under strong side illumination, which causes cast shadows, the shading-based refinement may fail, but for general unknown lighting (indoor ceiling or bright outdoor diffuse), it is able to produce good results for scenarios deemed challenging in previous works. Partial occlusions (e. g., hand, glasses, and hair) are difficult to handle with our dense optical flow optimization.

The inverse problem of estimating depth from a single image is far more challenging than in a multiview setting, and depending on the camera parameters, even notable depth changes of the head may lead to hardly perceivable differences in the projected image. Consequently, even though the tracked 3D geometry aligns well with the 2D video, there may be temporal noise in the estimated

[8] http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/

depth, which we filter out for the 3D visualizations. This limitation may stem from the use of a 2D PDM model and a 3D blendshape model that have a different dimensionality and expression range. In Chapter 7, we show an improved method that works towards a better coupling of these models for 3D pose estimation.

The dense tracking correction method proposed in Section 5.5 estimates a corrective 3D flow field to accurately track facial expressions, thus giving more personalization in the reconstruction. Due to the inherent depth ambiguity of our monocular setting, this corrective field is view dependent and cannot be correlated to the facial expressions performed by an actor to learn a personalized blendshape rig. In Chapter 7, we overcome this limitation by proposing a parametric corrective field based on harmonics functions that parametrize true 3D displacements and that can be regularized to control deformations in depth. Then, in Chapter 8 this parametric 3D correction field is learned as a function of the blendshape weights to create a rig with personalized expressions.

5.9 Summary

In this chapter, we have introduced a state-of-the-art method for monocular reconstruction of spatio-temporally coherent 3D facial performances. The proposed approach succeeds for scenes captured under uncontrolled and unknown lighting, and is able to reconstruct very long sequences, scenes showing very expressive facial gestures, and scenes exhibiting strong head motion. Compared to previously proposed model-based monocular approaches, it reconstructs facial meshes of very high detail and runs fully automatically, aside from a brief manual initialization step. It also fares very well in comparison to a recent state-of-the-art binocular facial performance capture method [Valgaerts et al. 2012b]. The proposed approach combines novel 2D/3D tracking and reconstruction methods, and estimates blendshape parameters that can be directly used by animators. Qualitative and quantitative results shown on several datasets demonstrate high tracking accuracy and overall good performance attained by the proposed method. We have also showcased its application to simple video editing tasks, such as appearance editing with virtual texture.

The results presented in this chapter advance the state of the art in monocular facial performance capture and show great potential for advanced video editing tasks. Next, in Chapter 6, we exploit the capabilities of our model-based method for video-realistic face retargeting, namely face expression adjustment for dubbing in movies.

Chapter 6

Model-based Face Retargeting: A Visual Dubbing Approach

(a) (b) (c) (d) (e)

Figure 6.1: The proposed visual dubbing approach modifies the lip motion of an actor in a target video (a) so that it aligns with a new audio track. The capture setup consists of a single video camera that films a dubber in a recording studio (b + c). The method transfers the dubber's mouth motion (d) to the actor and creates a new plausible video of the actor speaking in the dubbed language (e).

In this chapter, we build upon the high-quality model-based face capture approach presented in Chapter 5 to solve a challenging retargeting task that we refer to as visual dubbing. Traditional dubbing is a complex process used in the film industry that replaces the original actor's voice with that of a dubbing actor in a local language, such that the new audio stream adheres as best as possible to the actor's mouth motion in video. However, a result where the new audio and the original video are fully in sync is nearly impossible due to language differences. Thus, this chapter presents a visual dubbing approach that alters the mouth motion of a target actor in a video to match the new dubbed audio track (see Figure 6.1). The method and results presented in this chapter are based on [Garrido et al. 2015].

6.1 Introduction

Dubbing is the process of replacing the original voice of an actor in a video with a new one recorded off-camera in a studio. The new voice can reproduce the exact same original dialog, but with improved in-studio quality (this is referred to as post-synchronization, a.k.a "additional dialog recording"). However, in most cases the original actor's voice is substituted with that of a *dubbing actor* (or *dubber*) speaking in another language. Dubbing of foreign productions into the locally spoken language is common in countries where subtitling is not widely accepted, e. g., Germany, France and many Spanish speaking countries.

Dubbing has the advantage over subtitling that it does not draw the attention away from the action on screen. On the other hand, it has been shown that viewers are very sensitive to discrepancies between the auditory signal and the visual appearance of the face and lips during speech [Sumby and Pollack 1954]. In fact, audio-visual mismatches can drastically impair comprehension of the spoken language; hearing-impaired people in particular exploit this correlation even more [Owens and Blazek 1986; Summerfield 1992]. It is thus imperative that the dubbed language track is adjusted well to the visual performance. This requires an expensive and time consuming three-stage process performed by special production companies:

1. *Translation:* Certain mouth shapes are manually annotated in the video, such as the lip closure of the bilabial consonants /m/, /p/ and /b/. Then a transcript, which is semantically close to the original script and yet produces bilabials at roughly the same time, is made in the new language. Consequently, the translation may not be literal.

2. *Recording:* A dubber in a studio reads out the dubbed transcript in pace with the original performance. Even recording a single sentence may need several trials until alignment with the video is satisfactory.

3. *Editing:* The temporal alignment between the dubbed audio track and the mouth motion in the video is improved by manually time-shifting and skewing the new audio.

Despite the complexity of the pipeline, traditional dubbing is unable to produce dubbed voice tracks that match the mouth movements in the target video perfectly. The reason is that spoken words differ between languages, yielding different phoneme sequences and lip motions. Hearing and seeing different languages proves very distracting for many viewers [Sumby and Pollack 1954] and causes even stronger distraction for the hearing impaired who rely on lip reading [Owens and Blazek 1986].

This chapter introduces a system that visually alters the lip motion and the facial appearance of an actor in a video, so that it aligns with a dubbed foreign language voice. With this approach, we take a step towards reducing the strong visual discomfort caused by the audio-visual mismatch in traditional dubbing. Our method takes as input the actor's and the dubber's video as well as the dubbed language track, and then it employs state-of-the-art monocular facial performance capture to reconstruct both performances. This gives us parameters describing the facial performances based on a coarse blendshape model. Via inverse rendering, we additionally reconstruct the incident scene lighting in the target video, as well as the high-frequency surface geometry and dense albedo of the target actor. The captured dynamic 3D geometry of the actor is modified fully automatically by using a new space-time optimization method that retrieves a sequence of new facial shapes from the captured performance, such that it matches the blendshape sequence of the dubber, yet is temporally coherent, also in its fine-scale surface detail. A phonetic analysis of the dubbed audio finds salient

utterances, such as lip closures which are explicitly enforced in the synthesized performance. The synthesized face sequence is plausibly rendered and lit, after which the lower half of the face is seamlessly blended into the target video to yield the final result.

In summary, the main contributions are: 1) A visual dubbing system for video-realistic model-based resynthesis of detailed facial performances in monocular video that aligns the visual channel with a dubbed audio signal, 2) a spatio-temporal rearrangement strategy that utilizes the input facial performances and the dubbed audio channel to synthesize a new highly detailed 3D target performance, and 3) the reconstruction of a realistic target face albedo and the synthesis of a plausible mouth interior based on a geometric teeth proxy and inner mouth image warping.

The proposed method is one of the first to produce detailed, synthetically altered and relit facial performances of an actor's face. Our system generates visually plausible results which are compared against traditionally dubbed, unmodified video, both qualitatively and through a user study. Since the mouth region is completely synthesized in our approach, a perfect audio-video alignment is no longer required. Thus, the proposed approach simplifies the dubbing pipeline, since the translation into the foreign language can now stay closer to the original script.

6.2 Background: Visual Cues in Speech Perception

Visual cues, such as *visemes*, are essential for speech perception [Summerfield 1992], both for people with normal hearing [Owens and Blazek 1986] and in particular for hearing-impaired persons [Lesner and Kricos 1981]. In fact, under noise, one third of the speech information is conveyed visually through lip gestures [LeGoff et al. 1994] and a discrepancy between sound and facial motion clearly disturbs perception [Sumby and Pollack 1954]. The discrepancies between the visual and auditory cues can greatly change the sound perceived by the observer [McGurk and MacDonald 1976] and this may explain why many people dislike watching dubbed content [Kilborn 1993]. Taylor et al. [2012] report that a direct mapping from acoustic speech to facial deformation using visemes is simplistic and realistic synthesis of facial motion needs to model non-linear co-articulation effects [Slaney and Covell 2000]. The problem is that the statistical relationship between speech acoustics and facial configurations accounts for approximately 65% of the variance in facial motion [Yehia et al. 1998], and thus the speech signal alone is not sufficient to synthesize a full range of realistic facial expressions. In view of these findings, we build the mapping from the dubber to the actor primarily using the visual signal obtained through facial performance capture (see Section 6.4). We thus achieve audio-visual coherence implicitly, which it is reinforced by using the acoustic signal as a guide to enforce salient mouth motion events, like lip closures (see Section 6.6).

6.3 Overview

The proposed method takes as input two video recordings with sound[1]. The first recording is the original movie segment of the *actor* performing in the original language. We refer to this as the *target sequence* I_T, as it will be modified later. The second recording is the *dubbing sequence* I_D, showing the *dubber* reading a translation of the original text, which will serve as the source to synthesize a new target performance.

Our method uses the dubbed language track as the new voice track for the target sequence and

[1]Note that our approach does not need the actor's audio.

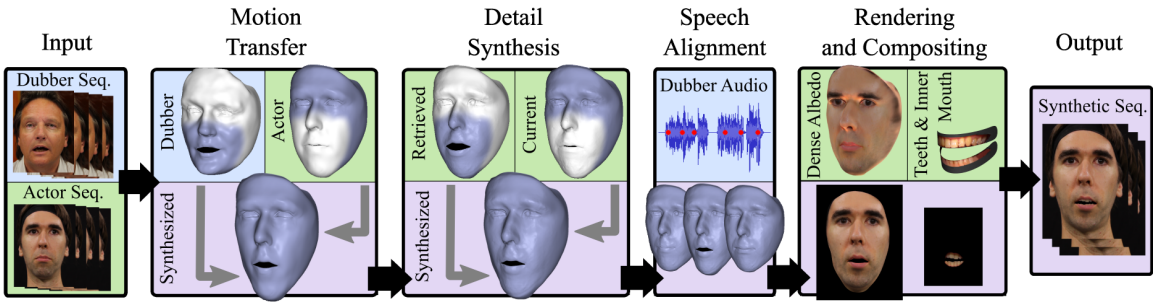


Figure 6.2: Method overview. Visual dubbing is performed in 4 main steps: Motion transfer (Section 6.4), detail synthesis (Section 6.5), speech alignment (Section 6.6), and rendering and compositing (Section 6.7).

modifies the mouth motion of the actor such that it matches the dubbed words. It does this while preserving the appearance and head pose of the actor, as well as the original background and scene lighting. We assume that the dubber reads his text roughly in pace with the actor's performance, but strict alignment of the dubbed language track with the actor's lip movements, as in traditional dubbing, is not necessary because a completely new synthesized performance is generated, which is in sync by construction. We further assume that the dubber is able to reenact the facial expressions of the actor well, i. e., the target and dubbing sequences bear a similar emotional content. The proposed method consists of four major steps, as shown in Figure 6.2:

S1 **Motion Transfer (Section 6.4):** The facial performances of the actor and the dubber are captured using a personalized blendshape model. The target lighting is estimated and high-frequency detail, such as wrinkles and folds, are captured. The blendshape weights pertaining to the mouth motion of the dubber are transferred to generate a new blendshape sequence for the actor.

S2 **Detail synthesis (Section 6.5):** Actor-specific high-frequency face detail is added to the synthesized blendshape sequence by globally searching for frames with similar detail in the target sequence. We only transfer detail in the lower face region around the mouth, preserving the original detail elsewhere.

S3 **Speech alignment (Section 6.6):** Lip closure is enforced by detecting bilabial consonants in the dubbed language track.

S4 **Rendering and compositing (Section 6.7):** By using the estimated target lighting and the dense skin reflectance of the actor, the synthesized face is rendered into the original video. The mouth interior is rendered separately and blended in with the target to produce the final composite.

In the remainder of this chapter $I_{\mathcal{T}}^t$ and $I_{\mathcal{D}}^t$ will denote the frame at time t in the target and dubbing sequence, with t running from 1 to the number of frames f. For simplicity, we assume that the target and dubbing sequence have the same number of frames and are temporally aligned such that corresponding spoken sentences coincide in time. This can be achieved as a preprocessing step or by recording the dubber in sync with the actor. The final result is the *synthesized sequence* I_S, showing the actor speaking in the dubbed language. More details on the different steps is provided as follows.

6.4 Motion Transfer

To capture the facial performances of the actor and the dubber, we employ the model-based facial performance capture approach presented in Chapter 5 that utilizes an underlying blendshape model and produces a sequence of space-time coherent face meshes with fine-scale skin detail. The parameters of the tracked blendshape model will be used to transfer the mouth motion from the dubber to the actor.

6.4.1 Monocular Facial Performance Capture

Both the actor's and the dubber's performance is captured using the method presented in Chapter 5, which uses a personalized blendshape model. This model is a prior on the face shape and describes a basis of variation in facial expressions:

$$\mathbf{e}(\beta_1, \ldots, \beta_k) = \mathbf{b}_0 + \sum_{j=1}^{k} \beta_j \mathbf{d}_j \ , \tag{6.1}$$

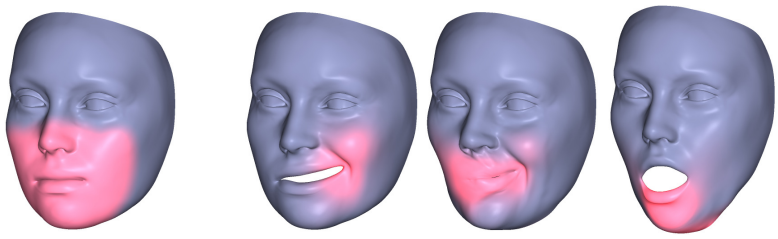
where $\mathbf{b}_0 \in \mathbb{R}^{3n}$ is a vector containing the n 3D vertex coordinates of the face at rest, $\mathbf{d}_j \in \mathbb{R}^{3n}$, $1 \leq j \leq k$, are the blendshape displacements at each vertex, and $\mathbf{e} \in \mathbb{R}^{3n}$ is the facial expression obtained by linearly combining the displacements using the *blendshape weights* $0 \leq \beta_j \leq 1$, $\forall j$. We create a personalized blendshape model of the actor and the dubber by registering a generic blendshape model to a static stereo reconstruction of the face at rest (see Section 5.3 for further details). Thus, the actor's blendshape model differs from that of the dubber's in face shape, but their k blendshapes correspond to the same canonical expressions and therefore have the same semantic meaning. For the models described in this section $k = 78$ and in the experiments we chose $n = 50000$.

As a brief recap, the monocular face capture approach first tracks the personalized blendshape model (3D rigid pose and blendshape weights) using $m = 66$ accurate and temporally stable facial landmarks, then it performs an out-of-space expression correction step using a temporally coherent dense motion field which better aligns the facial geometry to the face in the video, and finally adds fine-scale skin detail as a per-vertex surface displacement via shape-from-shading based refinement. The last step also estimates the scene lighting and a coarse, piece-wise constant approximation of the face albedo. The final result is a sequence of temporally coherent triangular face meshes $\mathcal{M}_{\mathcal{T}}^t$ for the target sequence and $\mathcal{M}_{\mathcal{D}}^t$ for the dubbing sequence, with $1 \leq t \leq f$.

It is important to remark that the dense expression correction step (see Section 5.5) performs a dense per-vertex mesh alignment that does not decouple rigid pose from facial motion nor does it provide an intuitive parametrization that could help perform editing tasks. As such, 3D deformations estimated by this step were not utilized in the motion transfer step nor were they employed to align the transferred mouth to the upper part of the face (i. e., we only used the rigid pose estimated with the 2D facial landmarks). This limitation is further discussed in Section 6.9.

6.4.2 Blendshape Weight-based Mouth Transfer

The blendshape model encodes most of the speech-related motion, such as the movement of the jaw, lips and cheeks, whereas the detail layer mainly encodes person-specific skin deformation, such as



(a) (b)

Figure 6.3: Mouth mask. *Left to right:* (a) The region of influence of the blendshapes responsible for the mouth motion, (b) three example blendshapes that activate the mouth, where the color encodes the magnitude of the displacement w. r. t. the rest pose.

emerging and shifting wrinkles. The blendshape models of the actor and dubber are derived from the same generic model and thus share the same semantic dimensions, including those related to speech. We can therefore make the actor utter the same words as the dubber by identifying the blendshape weights that activate the mouth, and by transferring the temporal curves of the blendshape weights that activate the mouth region from the dubber to the actor. As explained in Section 6.4.3, these activation curves will need further actor specific adjustment during transfer.

We manually identified the $l = 49$ blendshapes responsible for the mouth motion as those components that displace vertices on the jaw, lip or cheeks. We quantify a region of influence for these mouth blendshapes by assigning a value between 0 and 1 to each vertex, where 1 means highly affected by mouth motion and 0 not affected at all. These values are found by accumulating the l blendshape displacements at each vertex and mapping them to $[0,1]$, where 0 corresponds to zero displacement and 1 to the median displacement over all vertices. The obtained mask is depicted in Figure 6.3 and is used for detail synthesis and image blending (see Section 6.5 and Section 6.7). The mask is extended to include the nose tip, since it is often influenced by the mouth motion in practice.

The mouth motion of the dubber is transferred to the actor at a time t by combining the actor's blendshapes as follows:

$$\underbrace{\mathbf{e}_{\mathcal{S}}^{t}}_{\substack{\text{synthesized actor}\\ \text{expression}}} = \mathbf{b}_{0,\mathcal{T}} + \underbrace{\sum_{j=1}^{l} \beta_{\mathcal{D},j}^{t}\, \mathbf{d}_{\mathcal{T},j}}_{\substack{\text{captured dubber}\\ \text{expression}}} + \underbrace{\sum_{j=l+1}^{k} \beta_{\mathcal{T},j}^{t}\, \mathbf{d}_{\mathcal{T},j}}_{\substack{\text{captured actor}\\ \text{expression}}} . \tag{6.2}$$

Here, $\beta_{\mathcal{T},j}$ and $\beta_{\mathcal{D},j}$, $1 \leq j \leq k$, are the captured blendshape weights of the actor and the dubber, and $\mathbf{b}_{0,\mathcal{T}}$ and $\mathbf{d}_{\mathcal{T},j}$ denote the rest pose and the j-th blendshape of the actor. Note that the blendshapes are ordered such that mouth-related expressions come first in the model. The synthesized target expression $\mathbf{e}_{\mathcal{S}}^{t}$, $\forall t$, is identical to the original target expression, except in the mouth region shown in Figure 6.3 (a), where it is the same as the expression of the dubbing actor. The synthesized expression $\mathbf{e}_{\mathcal{S}}^{t}$ and the captured head pose can be used to build a sequence of synthetic, coarse face meshes $\mathcal{M}_{\mathcal{S}}^{t}$ for the actor, which exhibits the same mouth motion as the dubber. This is illustrated in Figure 6.5 for the example of Figure 6.1. Note that $\mathcal{M}_{\mathcal{S}}^{t}$ still lies within the blendshape space and therefore lacks any fine-scale detail, such as wrinkles and folds. This detail is necessary for a faithful rendering of the actor and will be added in Section 6.5.

6.4.3 Mouth Motion Correction

The blendshape weight-based transfer approach described in Equation 6.2 works well if the blendshape weight combinations for the actor and dubber have the same meaning. In practice, this is not guaranteed since both blendshape models are manually constructed from independently selected scans of a face at rest. As a result, they virtually share the same semantic dimensions, but do not necessarily agree on the rest pose, i. e., the two models span the same semantic space relative to the neutral pose but a blendshape, say $\mathbf{d}_j$, in the dubber's space may lie in a different place in the actor's space, resulting in a space misalignment as reported in [Theobald et al. 2009].

If there is a small systematic offset in the model origin, we can get an estimate of the true rest pose by selecting the blendshape weight combination that has the smallest Euclidean norm over all f captured frames, provided that there is at least one neutral expression in the sequence. This blendshape weight combination with minimum norm is then taken as the true model origin and is used to correct the transferred weights. To this end, we replace $\beta_{\mathcal{D},j}$ in Equation 6.2 by

$$\beta^*_{\mathcal{D},j} = \beta_{\mathcal{D},j} - \beta^{\min}_{\mathcal{D},j} + \beta^{\min}_{\mathcal{T},j} \quad \text{for} \quad 1 \le j \le l \ , \tag{6.3}$$

where

$$\beta^{\min}_{\mathcal{T}} = \underset{(\beta^1_{\mathcal{T}},\ldots,\beta^f_{\mathcal{T}})}{\arg\min} \parallel \beta^t_{\mathcal{T}} \parallel_2, \forall t \tag{6.4}$$

is the blendshape weight combination with the minimum Euclidean norm over all f target frames and $\beta^{\min}_{\mathcal{D}}$ has the same meaning for the dubbing sequence. We observed that this correction step significantly improved the quality of the expression transfer between different individuals. Note that the corrected weights $\beta^*_{\mathcal{D},j}$, $1 \le j \le l$ may lie outside the blendshape weight range. In practice, some weights were just slightly off the bounds, and therefore, it did not cause any visible artifact when transferring the dubber's mouth motion to the actor. Nonetheless, the corrected weights could be clamped in the range $[0,1]$ for consistency to allow animation artists to further alter the mouth motion, if desired.

6.5 Detail Synthesis

We add fine-scale skin detail to the synthesized target meshes $\mathcal{M}^t_{\mathcal{S}}$ by assuming that wrinkles and folds are correlated to the underlying facial expression, which in turn correlate to the blendshape weights. Detail in the top part of $\mathcal{M}^t_{\mathcal{S}}$ is not influenced by the blendshape weight transfer and can thus be assumed identical to that of the captured mesh $\mathcal{M}^t_{\mathcal{T}}$. Detail in the mouth region, on the other hand, changes under the effect of the new blendshape weights and must be synthesized appropriately. This detail has to be actor-specific and will be generated by first searching for similar expressions in the captured target sequence and then transferring the high-frequency detail layer from the retrieved target geometries.

6.5.1 Target Frame Retrieval: Energy Formulation

We wish to retrieve a captured target mesh $\mathcal{M}^{i(t)}_{\mathcal{T}}$ with a similar mouth expression and motion as the current synthesized mesh $\mathcal{M}^t_{\mathcal{S}}$. Here, $i(t) \in \{1,\ldots,f\}$ stands for the retrieved frame index in the

target sequence that corresponds to the current index t in the synthesized sequence. To this end, we look for similarities in the blendshape weights that drive the mouth motion of the mesh sequences $\mathcal{M}_T^t$ and $\mathcal{M}_S^t$.

Let β_j, $1 \leq j \leq l$, denote the set of blendshape weights that are responsible for the mouth motion, as identified in Section 6.4.2. Then we can represent the synthesized mouth expression at a frame t by the blendshape weight vector $B_S^t = \left(\beta_{S,1}^t, \ldots, \beta_{S,l}^t\right)^\top$ and the synthesized sequence of mouth expressions by $\mathbf{B}_S = (B_S^1, \ldots, B_S^f)$. Our retrieval problem aims at finding an optimal temporally-coherent rearrangement of target indices $(i(1), \ldots, i(f))$, such that the corresponding sequence of captured expressions $\hat{\mathbf{B}}_T = (B_T^{i(1)}, \ldots, B_T^{i(f)})$ is as close as possible to $\mathbf{B}_S$. This optimization problem can be formulated as:

$$\min_{(i(1),\ldots,i(f))} E\left(\hat{\mathbf{B}}_T, \mathbf{B}_S\right) \quad , \tag{6.5}$$

where E denotes a multi-objective function that measures the similarity of blendshape weights along with their change over time, and the adjacency of frames, described as follows.

Blendshape Weight Distance The similarity between a target and a synthesized mouth expression is computed as the L_2 norm of their difference. The index $i(t)$ of the target mesh, that is closest to the current synthetic mesh at frame t, has to minimize

$$d_{\mathrm{b}}(B_T^{i(t)}, B_S^t) = \left\| B_T^{i(t)} - B_S^t \right\|_2 \quad . \tag{6.6}$$

This distance measure is based on the assumption that, for a given person, face meshes with similar expression, and thus underlying blendshape weights, have similar skin detail.

Motion Distance To regularize the retrieval, we consider the change in expression over time, i. e., the difference between consecutive blendshape weights B^{t-1} and B^t. Given the expression change from $t-1$ to t in the synthesized sequence, we enforce that the currently retrieved blendshape weights $B_T^{i(t)}$ must undergo a similar change w. r. t. the previously retrieved weights $B_T^{i(t-1)}$. In other words, $i(t)$ and $i(t-1)$ have to minimize

$$d_{\mathrm{m}}(B_T^{i(t-1)}, B_T^{i(t)}, B_S^{t-1}, B_S^t) = \left\| (B_T^{i(t-1)} - B_T^{i(t)}) - (B_S^{t-1} - B_S^t) \right\|_2 \quad . \tag{6.7}$$

This measure assumes that similar changes in expression induce similar changes in skin detail. It is important to remark that the retrieved indices $i(t-1)$ and $i(t)$ do not have to be consecutive in the original target sequence, since the search is global.

Frame Distance Strong transitions in the retrieved detail are more likely if $i(t-1)$ and $i(t)$ lie far apart in the original target sequence. To enforce smoothly varying detail, the temporal distance of the retrieved neighboring indices is penalized as follows:

$$d_{\mathrm{f}}(i(t-1), i(t)) = 1 - \exp\left(-|i(t-1) - i(t)|\right) \quad . \tag{6.8}$$

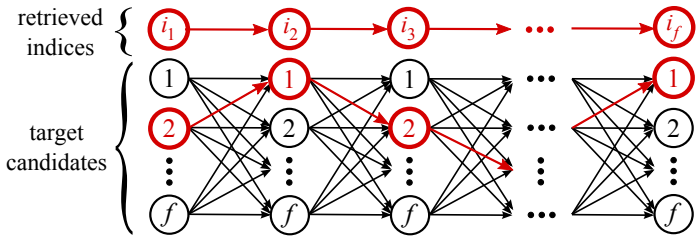


Figure 6.4: Shortest path in a graph of candidate indices.

This measure assumes that the captured facial details of close-by frames are more similar than those of distant frames.

6.5.2 Target Frame Retrieval: Energy Optimization

The optimal global rearrangement of target indices is then found by minimizing the energy in Equation 6.5, which is the weighted sum of the three distances over all frames:

$$E\left(\hat{\mathbf{B}}_T, \mathbf{B}_S\right) = w_\mathrm{b} \sum_{t=1}^{f} d_\mathrm{b}(B_T^{i(t)}, B_S^t) + w_\mathrm{m} \sum_{t=1}^{f} d_\mathrm{m}(B_T^{i(t-1)}, B_T^{i(t)}, B_S^{t-1}, B_S^t) + w_\mathrm{f} \sum_{t=1}^{f} d_\mathrm{f}(i(t-1), i(t)) \ , \tag{6.9}$$

where w_b, w_m, and w_f control the influence of each term.

A greedy approach could find the unknown indices sequentially by progressively retrieving the currently nearest one. A better solution that solves for the complete sequence $(i(1), \ldots, i(f))$ at once could be obtained by finding the shortest path in a weighted directed graph where each node represents a target index and each edge is weighted by the distances described above (see Figure 6.4). A solution can be found using Dijkstra's algorithm, but since the starting node is unknown, its complexity is $O(f^3)$ in the number of frames, which prohibits its use for long sequences. Instead, we can resort to methods based on hyper-heuristics [Burke et al. 2010] to arrive at an approximate solution that lies provably close to the global optimum. Hyper-heuristics are automated methods for selecting or generating local search operators to solve a hard combinatorial problem [Burke et al. 2013]. In our particular implementation, we define three local operators which independently minimize the three terms in Equation 6.9, as well as a fourth operator that randomly disrupts the local optimum at a random index location. The latter ensures that the algorithm can explore new solutions, avoiding stagnation in local minima. To guide the search for the optimal solution, we define a hyper-heuristic approach that adaptively selects these four operators by reinforcement learning, as originally proposed in [Garrido and Castro 2012].

Blendshape models can be overcomplete and multiple blendshape combinations may produce the same expression. We observed that different actors can activate distinct blendshapes when uttering the same words. As a consequence, facial expressions cannot be compared reliably using a distance between blendshape weights. This problem was overcome by performing Principal Component Analysis (PCA) on our blendshape model and replacing the blendshape weights in Equation 6.9 by the set of PCA weights that explains 99% of the mouth motion. Note that PCA does not change the face model; it only removes redundancy to make the frame retrieval more accurate. In the motion transfer step in Section 6.4, however, a blendshape representation is still preferred since the dimensions are spatially localized and easier to interpret [Lewis et al. 2014]. This provides

(a) (b) (c) (d) (e) (f)

Figure 6.5: Motion transfer and detail synthesis for the example of Figure 6.1. The facial performances of the actor (a) and the dubber (b) are captured, and the estimated mouth-related blendshape weights are transferred from the dubber to the actor, in this case making the actor open his mouth (c). Fine-scale facial detail from the captured mesh in the current frame (d) and detail from the captured mesh in the retrieved frame (e) are combined to produce a detailed synthetic mesh (f).

an extra level of control to the user who can globally scale the blendshape curves to modify the expressiveness. An example is illustrated in the supplementary video at the project website[2].

6.5.3 Analysis of Energy Terms and Parameter Tuning

To quantify the influence of the energy terms in Equation 6.9, we compared several retrieval results obtained with different values for the weights w_{b}, w_{m}, and w_{f}. To this end, a control sequence was also recorded for the experiment of Figure 6.8, in which the target actor is reading an English dubbing transcript under target conditions. The target and control sequences thus depict the same actor reciting the same dialog, both in German and in English. Based on the English audio, we selected the corresponding words in the dubbing sequence and control sequence which had a comparable timing, and identified 142 frames in which the visual utterance of the actor was identical to that of the dubber.

These 142 control frames were compared to the frames that were retrieved by our method from the German target sequence. If our frame retrieval is successful, the control frame and the retrieved target frame should depict the same utterance and look very similar. As a similarity measure, we used the average PSNR over all 142 frames. Small differences in the actor's pose were accounted for by warping the faces to a common reference pose. Retrieving the closest frames in time ($w_{\mathrm{b}} = w_{\mathrm{m}} = 0$) was least successful with an average PSNR of 22.0 dB. Retrieval purely based on the similarity of the PCA weights ($w_{\mathrm{m}} = w_{\mathrm{f}} = 0$) was more successful (28.0 dB), while adding the motion distance (28.2 dB) and the frame distance (28.6 dB) increased the similarity further. By using cross validation over a discrete set of parameters, we attained the best results by using the combination $w_{\mathrm{b}} = 1$, $w_{\mathrm{m}} = 10$ and $w_{\mathrm{f}} = 1000$, which was utilized in all of our experiments. Note that the control frames were not directly compared to our final synthesized images, since the rendering and the compositing can affect the PSNR adversely.

6.5.4 Detail Transfer

Once a sequence of target indices has been retrieved, we transfer the skin detail of the retrieved target mesh $\mathcal{M}_r^{i(t)}$ to the current synthesized mesh $\mathcal{M}_s^t$. The detail is added as a per-vertex displacement

[2] http://gvv.mpi-inf.mpg.de/projects/VisualDubbing/

Figure 6.6: Speech alignment. Lip closure (right) is enforced to improve audio-visual quality of the transferred mouth motion (left).

expressed in the local vertex coordinate frame. We only transfer new detail in the influence region of the mouth, given by the mask of Figure 6.3 (a). Outside this region we preserve the original detail of the captured mesh $\mathcal{M}_T^t$. At the mask boundary, we ensure a smooth transition between both detail layers using alpha blending.

Despite temporal regularization, the retrieved indices may still introduce slight jumps in the transferred detail (only the original ordering of target indices produces smooth detail over time, but does not resemble the dubbing performance). Thus, we temporally smooth out the transferred detail layer by filtering the displacements in a sliding Gaussian window of 5 frames. The detail transfer is illustrated in Figure 6.5.

6.6 Speech Alignment

We improve the synchronization of the lip motion and the dubbed audio by modifying the blendshape weights to enforce lip closures where needed. To determine the precise time instances of visually salient speech gestures, we analyze the audio of the dubbing sequence independently of the video stream. Since the content of the utterances spoken by the dubber is known, the audio was segmented into phonetic units using an automatic speech recognizer in forced-alignment mode [Young et al. 2006]. In the resulting phonetic segmentation, lip closure events are aligned by analyzing all instances of bilabial consonants /p/, /b/, and /m/. In many cases, the automatically determined segment boundaries are sufficient, but where reverberation or background noise in the recording affects the reliability of the automatic segmentation, the lip closure intervals were manually corrected using visual and acoustic cues in the phonetic analysis software Praat [Boersma and Weenink 2001]. The output is a sequence of time intervals associated with all speech-related lip closure events in the video sequences, at a precision far higher than can be achieved when analyzing only the dubber video footage.

The detected intervals are used to improve the timing of bilabial consonants in the synthesized video by forcing the blendshape weights responsible for lip closure to zero. To avoid jerky motion, enforcement is done in a small Gaussian window centered around the detected intervals (see Figure 6.6).

6.7 Rendering and Compositing

The synthesized meshes are rendered into the target camera using the estimated scene lighting and a per-vertex estimate of the skin reflectance. In the last step, the mouth cavity and the teeth are then rendered and combined to produce the final composite.

6.7.1 Rendering the Synthesized Geometry

Although complex light transport mechanisms (e. g., such as subsurface scattering) influence the perceived skin color, we assume pure Lambertian skin reflectance, which is sufficient under most conditions. To this end, we use the following formulation of the rendering equation:

$$\mathcal{B}(\mathbf{v},\omega) = \mathbf{c}(\mathbf{v}) \int_{\Omega} L(\mathbf{v},\omega) V(\mathbf{v}) \max(\omega \cdot \mathbf{n}(\mathbf{v}), 0)\, \mathrm{d}\omega \quad , \tag{6.10}$$

where $\mathcal{B}(\mathbf{v},\omega)$ is the irradiance at vertex $\mathbf{v} \in \mathbb{R}^3$ from an incoming light direction ω sampled on the hemisphere Ω, $\mathbf{c} \in \mathbb{R}^3$ denotes the skin albedo at vertex $\mathbf{v}$, $\mathbf{n} \in \mathbb{R}^3$ represents the surface normal at vertex $\mathbf{v}$, and $V \in \{0,1\}$ is the vertex visibility (please refer to Section 2.2.2 for further details).

As mentioned in Section 6.4.1, the monocular performance capture method presented in Chapter 5 estimates a coarse, piece-wise constant albedo of the actor's skin albedo, as well as the scene lighting $L \in \mathbb{R}$ in the target scene (here represented as white illumination). However, the coarse albedo is insufficient for a convincing rendering of the actor and we require a per-vertex albedo $\mathbf{c}(\mathbf{v})$ instead. We estimate the dense skin albedo by projecting each vertex $\mathbf{v}$ of the captured mesh $\mathcal{M}_T^t$ into the target frame I_T^t and assigning the intensity to $\mathcal{B}(\mathbf{v},\omega)$ in Equation 6.10. Dividing the irradiance by the integral on the right then gives us an estimate of $\mathbf{c}(\mathbf{v})$. We can then render the synthetic mesh by solving the rendering equation for each vertex of $\mathcal{M}_S^t$.

If the dense albedo is estimated for each frame independently, it may suffer from small imprecisions in the captured face geometry and lead to undesirable intensity changes in the rendered images. To avoid this, we assume that the albedo is constant over time and estimate a single value in each vertex via a least squares fit over all captured meshes. To improve spatial sampling, albedo computation and rendering are performed on upsampled versions of the face meshes ($n = 200000$).

6.7.2 Teeth, Inner Mouth and Final Composite

As illustrated in Figure 6.7 (d) the rendered face lacks teeth and a mouth cavity. For the upper and lower teeth, we create a 3D teeth proxy consisting of two billboards that are attached to the blendshape model (see Figure 6.7 (a)) and move in accordance with the face under the control of the blendshape weights. The billboards are colored with a static texture (see Figure 6.7 (c)) of the target frame in which the teeth are visible. The inner mouth is created by warping a single image of the mouth cavity (see Figure 6.7 (b)) using the facial landmarks obtained from the synthesized facial performance. The brightness of the teeth and inner mouth is uniformly adjusted according to the degree of mouth opening to create a realistic shading effect.

The warped inner mouth, rendered teeth and synthetic face layers are sequentially rendered and blended in with the target image by feathering around the boundaries to assure a smooth transition, as shown in Figure 6.7 (e). We only blend the synthesized face inside the projection of the mask of

(a) (b) (c) (d) (e)

Figure 6.7: Rendering and compositing. The textured 3D teeth proxy is anchored to the blendshape model to simulate the opening/closing (a). The inner mouth (b), the upper and lower teeth billboards proxies (c), and the synthesized face (d) are rendered on top of each other to produce the final composite (e).

Figure 6.3 (a), while preserving the original face elsewhere. The result is the synthesized sequence I_S^t.

6.8 Experiments

We applied our method to three target sequences of German-speaking actors recorded under constant, unknown illumination. A dubbing studio[3] translated the original German scripts and recorded a new English language track for each sequence using a professional dubber. The dubber was filmed in the studio with the setup of Figure 6.1. The central camera is used for performance capture, while the two satellite cameras are only used for the 3D reconstruction needed for the blendshape creation. All videos where shot with an SLR camera at 25 fps in HD quality. The German audio was recorded with a USB microphone and the English audio with the dubbing studio equipment.

As the dubbing results reported in this section contain audio content, the reader is strongly advised to watch the supplemental video at the project website[4]. Besides, relevant comparisons described in Section 6.8.2 are also better appreciated in video.

6.8.1 Results

Figure 6.8 presents our result attained on the first sequence for a target actor reciting a dialog of a movie in German. This sequence is 1.5 min long and the actor remained mostly still while speaking, which illustrates the quality that our method can achieve in ideal conditions. The upper row in the figure shows example frames from the target sequence, whereas the middle row shows the corresponding frames from the English dubber sequence. These are assumed to be correctly aligned in time such that the English and German sentences overlap. As most professional dubbing studios record single sentences in separate takes, this alignment had to be performed manually (only at the beginning of the takes). The bottom row of the figure shows the corresponding synthesized results. The mouth motion, the actor appearance, and the mouth interior are plausibly synthesized. The supplementary video available at the project website further demonstrates that the synthesized mouth motion matches the dubbed audio track well.

[3] SPEEECH Audiolingual Labs, www.speeech.de

[4] http://gvv.mpi-inf.mpg.de/projects/VisualDubbing/

Figure 6.8: Dubbing results - first sequence showing an actor reciting a passage of a movie. *Top to bottom row:* Target actor, dubber and synthesized result.

Figure 6.9: Dubbing results - second sequence depicting a scene of a passion play. *Top to bottom row:* Target actor, dubber and synthesized result.

Figure 6.10: Dubbing results - third sequence showing an actor being interviewed. *Top to bottom row:* Target actor, dubber and synthesized result.

Another result obtained on a second sequence for a different target actor performing a scene of a passion play can be seen in Figure 6.9. This sequence is challenging due to the fast head motion and the expressive facial gestures. The figure shows that the new mouth motion and appearance are plausibly generated and much of the emotional content is preserved, which demonstrates that our method is capable of dealing with fast and expressive motion. Finally, Figure 6.10 shows a third sequence where the same actor is answering questions from an interviewer. This video attempts to simulate a television interview where the spoken lines are spontaneous and not scripted beforehand. Also for this result, the expressions of the actor, including laughter and pondering gestures, are preserved well.

6.8.2 Validations

User Study We conducted a web-based user study in which we asked users with an understanding knowledge of English to compare the results shown in Figures 6.8–6.10 with those obtained using traditional dubbing. Note that the comparison was done side by side in a random order. The traditionally dubbed results were provided by the same dubbing studio that recorded the dubbing actor and dubbed the German language track into English. These videos were generated by taking the original German target videos, removing the original audio, and adding the dubbed language track in English. Note that the dubbed language track was further altered (i. e., manually time-shifted and skewed) by one of the experts in the dubbing studio to improve the overall audio-visual alignment, thus creating high-quality professional videos (please refer to the second supplementary video at the project website for these results). The results corresponding to the same sequence were equally long and their lengths, as well as other features, including the amount of head motion is

Table 6.1: User study. Length and main features of the sequences used in the survey.

Sequence	length (seconds)	head motion	head orientation
Movie dialog (Figure 6.8)	30	negligible	frontal
Passion play (Figure 6.9)	20	strong/fast	frontal/non-frontal
Interview (Figure 6.10)	30	mild	mostly non-frontal

described in Table 6.1.

To quantify the quality of the dubbing, in the webpage we attached to the results a questionnaire that evaluated the overall audio-visual experience, including overall viewing discomfort and how natural the video-audio combination was perceived by the user. To be more precise, we included a Likert scale that ranged from 0 to 5, where 0 means a really bad overall visual-audio experience, whereas 5 means a very good experience. To collect some statistics about the user preference, we also asked the users to give their preference for one of the two approaches. An optional comment box was also included. The web-based user study was sent to 45 participants from different places around the world, including countries were dubbing is a common practice (Germany and France) and also countries were it is not (UK, USA and Chile).

Table 6.2 summarizes the overall scores assigned to each sequence, as well as user preferences. Over all three sequences, traditional dubbing received an average score of 3.2, while our visual dubbing system received a score of 2.7. Overall, 35% of the respondents said they felt more comfortable watching the visually modified video. These scores seem low at first, but actually indicate a big step ahead in solving this extremely difficult problem. The human eye is tuned to the slightest visual artifact in a rendered face and it is very hard for an automatic system to produce visually plausible results that do not fall in the uncanny valley, especially in a side-by-side comparison against real video. Despite the professional quality, traditional dubbing was not favored by everyone. In fact, the visually modified result of Figure 6.8 was preferred by 47% of the users and we believe this shows considerable progress towards a system that can replace facial performances in video. The same result received an absolute score of 2.7, which is only slightly less than the 2.9 score of traditional dubbing. Overall, the result shown in Figure 6.10 received the highest score of 3.0.

We additionally performed the ANOVA F-test to find the statistical significance of the scores obtained in the user study. The p-values were ~ 0.4, 0.001, and 0.006 for the results of Figure 6.8, Figure 6.9, Figure 6.10, respectively. This means that two out of three experiments were statistically significant, as their p-value falls below 0.01, i. e., the random sampling error in the user study is less than 1%. The high p-value of the experiment related to the result of Figure 6.8 can be ascribed to the high standard deviations and the tied scores compared to the others, meaning that more samples would be needed to have conclusive statistics. However, we believe that the scores for this sequence illustrate a trend towards equal appreciation of our results and those of the studio. In general, the variability of the scores can be explained by two main criteria that the users found very relevant: lip-sync and expressiveness/realism. Some survey respondents preferred good lip-sync to out-of-sync expressive faces, but also the other way round. Some of the comments left by the participants include: “Sometimes exaggerated expression is better”, “I voted for the videos where the sound-image synchronization was better”, “In my opinion, it is not just about making the mouth move in line with the audio”, “I feel that the synchronization by itself always looked very good”.

Table 6.2: User study. Scores given by the survey respondents to the results obtained by traditional dubbing and our approach, as well as their overall preference.

Sequence	Traditional dubbing		Our approach	
	Score	Preference	Score	Preference
Movie dialog (Figure 6.8)	2.9 ± 1.2	53%	2.7 ± 1.1	47%
Passion play (Figure 6.9)	3.0 ± 1.2	73%	2.3 ± 0.9	27%
Interview (Figure 6.10)	3.6 ± 1.1	70%	3.0 ± 0.9	30%
Overall	3.2 ± 1.2	65%	2.7 ± 1.0	35%

Figure 6.11: Renderings with (left) and without synthesized skin details (right). Without added detail the face looks over smoothed, and therefore non-realistic.

Rendering Figure 6.11 demonstrates the importance of facial detail synthesis for photo-realistic rendering by comparing our result with a system that renders the face using a blendshape model without fine-scale detail. This corresponds to facial replacement/reenactment techniques that use a coarse 3D parametric model without a detail layer [Dale et al. 2011; Thies et al. 2016]. Compared to the proposed method, important skin features, such as dimples, are hardly visible without a geometric detail layer and realistic shading effects on the chin and upper lip are also missing. The supplementary video at the project website also compares alternative strategies to create the inner mouth, showing that the proposed compositing based on multiple layers achieves the best results.

Comparison to Image-based Methods To demonstrate that our 3D model-based approach outperforms 2D image-based approaches, we compare the proposed visual dubbing approach to a modified version which does not produce the final composites by rendering a synthesized 3D geometry, but by reordering the frames of the target actor and then applying non-rigid 2D warping, as presented in Chapter 4. Such a method is then similar to a purely image-based technique, like Video Rewrite [Bregler et al. 1997], but with better image warping. The method presented in Chapter 4, however, creates a new synthesized sequence with a facial performance that is close to that of the dubber, but it warps the actor into the dubbing sequence (instead of the original target sequence) and mixes the identities of the dubber and the actor, which is not suitable for the dubbing scenario described in this chapter.

We can design an image-based approach that is suitable for the scenario at hand by retrieving target frames in the actor sequence that match the dubber's expressions (see Section 4.4 for further details), but warping the face region of the retrieved target frames back into the original target sequence. To assist the warping, we use the synthesized facial landmarks that are provided by the motion transfer step in Section 6.4. These landmarks correspond to the actor's face in the target sequence, but move in accordance to the dubber's speech. For the shape/texture warping, we can use the same non-rigid 2D mapping described in Section 4.5. The resulting strategy is image-based and ensures that the mouth motion in the warped frames moves in pace with the dubber's mouth motion while being

Figure 6.12: Final composite using the proposed model-based approach (top), final composite obtained by the image-based approach (middle), and the corresponding frames from the dubbing sequence (bottom).

correctly aligned to the actor's face in the original target sequence.

Figure 6.12 shows some of the results obtained by our visual dubbing approach and by the image-based approach on the sequence of Figure 6.9. Note that the image-based approach replaces the complete inner face, while the proposed method only replaces the lower part of the face. The image-based results can suffer from ghosting artifacts (third column), may not always be in pace with the dubber's performance (second and fifth column), and may even struggle with strong head motion as a result of unrealistic face warping (third column). These and other issues, such as the temporal alignment of the mouth region and the temporal resolution, can more clearly be seen in the supplementary video at the project website. This demonstrates that our model-based approach produces synthesized sequences of overall higher quality in terms of the spatio-temporal resolution and can deal well with challenging sequences that exhibit fast and strong head motion, where image-based approaches normally have trouble.

6.9 Discussion and Limitations

The proposed approach takes a notable step ahead over previous facial expression transfer or facial video modification approaches. Unlike video rewrite [Bregler et al. 1997] or model-based replacement methods that mix identities [Dale et al. 2011], we can synthesize results when target and dubbing actor are *different*, which is essential for any practical application. The use of an accurate parametric face model, along with detailed lighting and albedo information enables photo-realistic synthesis of face appearance, even on long videos with moderate out-of-plane head motion. As shown in the experiments, the proposed 3D model-based resynthesis approach bears several advantages over purely or model-assisted image-based methods (see Chapter 4), which often exhibit

ghosting artifacts or temporal aliasing, merely show results without compositing, and can only handle marginal out-of-plane head motion, as already discussed in Section 3.5.2.

Our visual dubbing approach also takes a big step towards easing and streamlining the workflow of traditional dubbing: We no longer require a translation of the original text that (perfectly) matches the visual utterances in the target video on a viseme level. Since we resynthesize the mouth motion entirely, the translation can be more free. Furthermore, the proposed method relies on very little manual preprocessing, most notably the creation of the blendshape model and teeth proxy (see further discussion below); otherwise, it is fully automatic and can be integrated into an industrial pipeline.

Since the proposed method is the first step towards solving a challenging goal, it has several limitations. First, a static 3D reconstruction of the actor's face is required to build a coarse personalized blendshape model, but it may not always be available for every actor, especially from vintage movies. Automatic reconstruction of blendshape models from video is first addressed in Chapter 7, and further extended in Chapter 8 by learning from the captured data personalized, fully-controllable face rigs that can synthesize person-specific expressions and fine-scale skin details.

Regarding the mouth motion transfer step, even though the proposed approach corrects the motion curves transferred to the actor's model to account for differences in identity (face shape), it still imposes the dubber's idiosyncrasies onto the target actor, resulting in synthesized sequences that reflect the characteristics of the dubber rather than of the original actor. For instance, in the tracked sequences we measured an asymmetry in the blendshape weights of the dubber as part of his natural way of speaking and this asymmetry was reproduced in the actor (please refer to the result of Figure 6.8 in the supplementary video available at the project website). This problem was also reported in [Theobald et al. 2009]. Even more sophisticated expression cloning methods that rely on model-specific priors to constraint the range of plausible expressions [Seol et al. 2012] would still transfer dubber characteristics if no direct control from a user is provided. To deal with this problem, certain aspects and weights could be manually controlled to achieve the desired amount of expressiveness, but this may require certain user expertise. Alternatively, differences between the two actors could be learned in order to achieve a certain style (see Section 3.4.2 for more details); however, this requires source and target training examples with semantically similar expressions. In Chapter 8, we tackle this problem by learning a face rig that couples generic blendshapes to detailed reconstructions to generate person-specific shape details and expressions. This way, new performances that *preserve actor's characteristics* could be synthesized by just transferring standard blendshapes captured from the dubber's performance.

As mentioned in Section 6.4.1, the monocular face capture approach presented in Chapter 5 performs an out-of-space blendshape deformation to improve alignment. However, this correction step simultaneously improves expression, shape and rigid pose (all coupled together); therefore, the corrective deformation field cannot be directly applied for expression transfer nor can it be used for aligning the synthesized faces. Hence, such alignment step was not employed for transfer, but could certainly help produce higher fidelity results. In Chapters 7–8, we propose a multilayer parametric model that can effectively decouple rigid pose from person-specific deformations, and also a face rig that automatically generates such deformations from blendshape weights, as mentioned above. We feel confident that such improvements will contribute to attaining better results in this dubbing scenario.

For the application at hand only the mouth region is replaced and not the full facial expression, which may not convey all of the visual information in the speech. For plausible results, the dubber is expected to play the same routine as the actor with similar emotional content, which is mostly

fulfilled in practice. However, there may be facial regions where a match between the new mouth motion and the original video is challenged, e. g., the larynx in our results does not move according to the dubbed audio.

We only detect lip closure and opening events from the audio track. As audio cues provide a precision far higher than video only, more complex information, such as triphones or alike could be extracted and used to train for instance a hidden Markov model [Anderson et al. 2013a; Brand 1999]. This may further improve results. Furthermore, we compute an average albedo, which can be blurred if the correspondences given by the monocular tracker are not accurate over time. The current lighting model may be challenged in scenes with strong and sudden light changes, and the current monocular tracking may fail in extreme facial poses, e. g., completely lateral views. In this chapter, a simple approach that uses a textured teeth proxy is proposed for synthesizing part of the mouth interior, but the rendered teeth may not always look realistic. Some of the limitations mentioned above, especially regarding the digitization of the mouth cavity, will be further discussed in Chapter 10.

6.10 Summary

In this chapter, we have presented one of the first automatic solutions for transferring expressions between two different real-life actors and rendering photo-realistic, plausible mouth motion in an existing video that visually correlates to a dubbed audio track in a different language. The approach is based on highly detailed monocular 3D face reconstruction, as well as lighting and albedo estimation. New 3D mouth performances are synthesized by using a new parameter-based motion transfer step between dubbing and target actor, and a space-time retrieval method that synthesizes plausible high-frequency shape detail. The synthesized results, including the interior of the mouth, are photo-realistically rendered and attention is paid to a proper synchronization of the mouth motion with salient utterances in the audio track. Resynthesizing facial motion at video quality is extremely challenging as our perception is attuned to the slightest inaccuracies. Qualitative comparisons and a user study conducted on several individuals from different countries have shown that the proposed method can create plausible results and that we have taken an important step towards solving this challenging problem.

The algorithms proposed thus far have shown promising results in real-life scenarios and advance the state of the art in monocular face capture and video-based editing in semi-controlled monocular setups (i. e., monocular cameras with known intrinsics and available 3D reconstruction of the actor's face). The reconstructed models still lack an intuitive parametrization of person-specific expressions and details to allow digital artists to perform advanced animation and editing tasks at much higher granularity. Digitization of photo-realistic, fully-controllable and highly-personalized 3D face avatars in unconstrained monocular setups at much higher degrees of detail and personalized control is addressed next in Chapter 7 (and also in Chapter 8).

Chapter 7

Multilayer Model-based Face Capture in Unconstrained Setups

Figure 7.1: Result obtained by the proposed approach on a video downloaded from YouTube. *Top:* Input video (https://youtu.be/d-VaUaTF3_k). *Bottom:* Reconstructed high-quality, personalized 3D model that captures: Coarse-scale face geometry and expressions, medium-scale person-specific idiosyncrasies, and fine-scale skin detail – all directly from monocular video.

Chapters 5–6 presented robust model-based methods for capturing and transferring detailed facial performances. These methods, however, are unable to track 3D faces in completely unconstrained 2D videos (e. g., vintage footage), and estimate/parametrize person-specific mid-scale deformations. This chapter presents a fully automatic multilayer model-based approach for capturing arbitrary performances at multiple levels of details from 2D video with unknown camera and lighting setups where we do not have access to the actor's face geometry, e. g., internet videos (see Figure 7.1). The method and results presented in this chapter are based on [Garrido et al. 2016a].

7.1 Introduction

Creating photo-realistic face animations of virtual actors in movies and games is a very challenging task, since human perception is attuned to detecting even small inaccuracies in facial appearance and expression. Hence, animation artists strive to construct and animate high-quality controllable 3D models (or face rigs), especially when photorealism is the goal. The animation process often requires a 3D face scan of the actor, as well as detailed blendshapes captured in complex setups, which are normally retouched and then manually combined by artists. To simplify this complex pipeline, researchers have developed different face capture methods in an attempt to automatize most of its steps (see details in Section 3.1). Some methods can track detailed blendshape models, while others can reconstruct detailed dynamic face geometry and appearance from scratch, using either complex scanning systems or multiview camera setups under controlled illumination (Section 3.1.1). Recently, performance capture methods have further been extended to work with RGB-D sensors (Section 3.1.2) or even just RGB video filmed under general conditions (Section 3.1.3).

Despite the high-level of detail and tracking accuracy achieved by monocular approaches, most of them assume certain knowledge about the scene or require a 3D neutral model of the actor's face. Besides, they neither estimate nor parametrize personalized mid-scale deformations, such as person-specific smiles, nose shapes, etc. Capturing such deformations not only contributes to a better tracking, but also help decouple fine-scale transient details from true facial motion. This will be quite beneficial when learning and especially editing face rigs, as demonstrated in Chapter 8.

In view of the current limitations, this chapter introduces a novel multilayer model-based approach for capturing fully parametrized 3D models from unconstrained performances where no knowledge about the scene is given, e. g., arbitrary videos downloaded from the Internet. At the heart of this approach is a new parametric face prior that jointly encodes plausible appearance and shape changes. The appearance is modeled assuming Lambertian reflectance, whereas the shape is encoded by a subspace of facial identity, person-specific expression variation and dynamics, and fine-scale skin detail formation. Contrary to the method presented in Chapter 5 that focuses on acquiring accurate 3D face geometry without detailed personalization, we capture a detailed parametric 3D face model which is gradually refined at multiple layers to model the specifics of a person. At the coarsest layer, shape identity is parametrized using a principal component model and facial expressions are represented with a generic blendshape model. Person-specific idiosyncrasies in expression and identity not modeled in this generic space are captured by a second layer using medium-scale corrective shapes. A generative fine-scale detail model reconstructed over the face surface constitutes the final most detailed layer. The parameters of this multilayer model are personalized to an actor's video by utilizing a new analysis-by-synthesis fitting approach to recover the coarse and medium layers, as well as a shading-based refinement approach under general lighting to extract fine-scale detail. The output of our algorithm is the personalized 3D face model (including all its shape-related parameters), a detailed face albedo map, and an estimate of the scene lighting and camera's parameters.

The method proposed in this chapter captures detailed, personalized models from arbitrary monocular video of actors, even from vintage footage, for which it would be impossible to automatically capture the performance by any other standard means. It is important to stress that our approach does not require manual intervention during fitting as in [Alexander et al. 2010; Weise et al. 2011], nor does it need dense static 3D face geometry captured in a pre-processing step [Fyffe et al. 2014; Ichim et al. 2015; Valgaerts et al. 2012b].

In summary the main contributions are: 1) A new automatic model-based approach for capturing detailed personalized models from unconstrained monocular video, 2) a new multilayer paramet-

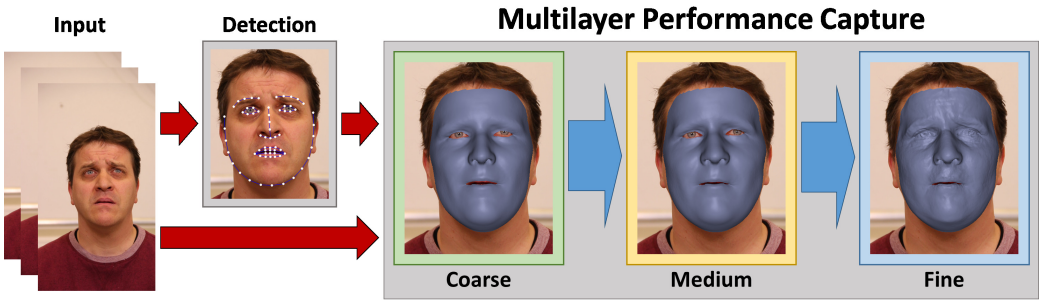


Figure 7.2: Overview. Given an unconstrained monocular video and accurately detected 2D landmarks, we first reconstruct a coarse model of the actor (identity and expressions) $\mathcal{M}^C$ and also person-specific medium-scale deformations $\mathcal{M}^M$ (actor's characteristics) – all based on a novel fitting energy. Finally, fine-scale details $\mathcal{M}^F$ are estimated in a shape-from-shading framework.

ric face representation in shape to reconstruct and represent 3D facial surface at different levels of detail, and 3) a unified novel fitting approach based on inverse rendering that leverages both color cues and sparse 2D landmarks to reconstruct the facial geometry at the coarse and medium layer. Qualitative and quantitative results show that our multilayer face capture approach compares favorably to alternative monocular and multiview methods in terms of reconstruction accuracy (see Section 7.7.2).

7.2 Overview

The proposed method takes as input an unconstrained monocular video $\mathcal{F} = \{f_1, \ldots, f_T\}$ as well as 2D facial landmarks $\mathcal{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_T\}$ tracked throughout the sequence[1], as shown in Figure 7.2. Note that T represents the total number of frames. The video can be recorded indoors, outdoors, or downloaded from the Internet (e. g., vintage movie or YouTube video); therefore, the scene information and 3D geometry of the actor's face are generally unknown. To reconstruct the actor's face shape and appearance in the video, we propose an approach that inverts the image formation process to model all relevant scene components, including camera parameters, scene lighting, skin reflectance and dynamic 3D face geometry parametrized on multiple personalization layers. This is performed in three main steps, as follows:

S0 **Multilayer Personalized 3D Face Prior Creation (Section 7.3):** We construct an adaptive parametric 3D face prior that models the complete image formation process on a simple full perspective camera projection model, as described in Section 2.2.1. This prior consists of the camera's intrinsics and extrinsics, the scene lighting, and a multilayer parametric 3D face model which encodes actor-specific facial appearance and geometry, as well as motion on three different layers: coarse-scale shape $\mathcal{M}^C$, medium-scale corrective shapes $\mathcal{M}^M$, and fine-scale skin detail $\mathcal{M}^F$ on the wrinkle level. This prior is modeled only once and its parameters are updated to fit the observed face in the image.

S1 **Coarse- and Medium-scale Layer Reconstruction (Section 7.4):** We first track a generic actor model from video by using a novel tracking energy that jointly optimizes for facial

[1]Refer to Section 4.3 for further details on 2D facial landmark tracking.

shape, expression and illumination parameters, such that a photometric and feature consistency measure is maximized. In this analysis-by-synthesis process, the camera's parameters are also estimated. Starting from this initial coarse shape and motion estimate $\mathcal{M}^C$, the quality of the fit is further improved based on linear person-specific correctives, thereby yielding a medium-scale corrective layer $\mathcal{M}^M$.

S2 **Fine-scale Layer Reconstruction (Section 7.5):** In a final step, we utilize inverse rendering to solve for a wrinkle-level detail layer $\mathcal{M}^F$ that optimally agrees with temporal changes in shading cues observed in the input image sequence.

Section 7.6 provides a comprehensive algorithmic description of the multi-step optimization strategy that was adopted to estimate all the scene parameters and personalization layers. The output of the proposed method is the camera's parameters, the scene lighting and a multilayer personalized 3D face model of the actor $\mathcal{M}_t = \{\mathcal{M}_t^C, \mathcal{M}_t^M, \mathcal{M}_t^F\}$ at each frame f_t, $\forall t$, including all extracted shape parameters in the different layers, as well as an albedo map of the actor's face.

7.3 Multilayer Personalized 3D Face Prior

Our reconstruction process inverts the image formation in the scene and recovers the camera's parameters, the scene lighting, and the multilayer face model that comprises the actor's appearance, identity (shape) and expression (deformation) parameters. Facial identity and expression variation is parametrized on three different layers, as shown in Figure 7.3: a coarse-scale linear parametrization of identity and expression, medium-scale corrective shapes based on manifold harmonics and a fine-scale detail layer at the wrinkle level represented as triangle deformations. In the following, these components are explained in more detail.

7.3.1 Camera Parametrization

To project the personalized parametric 3D model onto the image, we adopt a standard perspective pinhole camera model with camera space position $\mathbf{t} \in \mathbb{R}^3$ and orientation $\mathbf{R} \in \mathrm{SO}(3)$, where the world coordinate system is centered at $[0,0,0]^\top$. Note that the coordinate system of the parametric 3D model is assumed to be aligned with the world space. Hence, $\mathcal{C}(\mathbf{v}) = \mathbf{R}\mathbf{v} + \mathbf{t}$ maps a world space point $\mathbf{v} \in \mathbb{R}^3$ to the camera's local coordinate frame. An image of the parametric model in 3D world space is then formed by projecting each surface point $\mathbf{v}$ of the model onto the camera's 2D image plane, as follows:

$$\hat{K}(\Pi\mathcal{C}(\mathbf{v})) = \hat{\Pi}\mathcal{C}(\mathbf{v}) \ , \tag{7.1}$$

where $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ denotes a non-linear operator in homogeneous coordinates that performs perspective projection and $\hat{K} \in \mathbb{R}^{2x3}$ is the matrix containing the camera's intrinsics parameters in non-homogeneous coordinates. A more detailed description of these operators can be found in Section 2.2.1. Hence $\hat{\Pi} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ corresponds to the camera's full perspective transformation that converts a surface point into an image point.

In a pre-processing step, the camera's full perspective transformation $\hat{\Pi}$ is obtained by estimating the optimal intrinsic camera parameters. To be more precise, the focal length, rigid-head pose and the actor-specific shape parameters are jointly optimized based on the sparse set of accurately

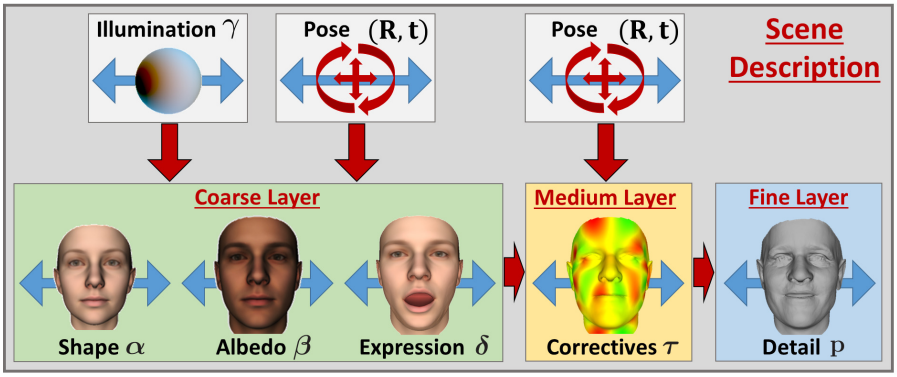


Figure 7.3: Scene description. A novel multilayer person-specific model is used to parametrize the identity, facial motion, person-specific idiosyncrasies and fine-scale details of the actor's face on monocular video input. In addition, extrinsic camera parameters and the scene lighting are also extracted.

tracked 2D facial landmarks (see Section 4.3) over the first 100 frames of the input video sequence. Note that the principal point is assumed to lie at the image center for the sake of simplicity, but it could also be included in the optimization. However, this may result in overall less accurate parameter estimates (please remember that this is already an ill-posed problem and more parameters will add more uncertainty in the estimation).

7.3.2 Lighting and Appearance Model

Here, we assume a pure *Lambertian* skin reflectance model as in Chapters 5–6 and later works [Ichim et al. 2015; Shi et al. 2014; Suwajanakorn et al. 2014]. This is a simplification of true skin reflectance that offers a good trade-off between complexity and quality of the obtained results. Since the scene is assumed to be purely Lambertian, the global illumination in the scene is represented using a spherical environment based on *spherical harmonics* (SH) basis functions [Müller 1966]. In spirit of Ramamoorthi and Hanrahan [2001], the first $B = 3$ SH bands are used here to represent the outgoing lighting reflected at a surface point with surface orientation $\mathbf{n}$ and skin albedo $\mathbf{c}$. Hence, the irradiance at that point can be parametrized in terms of the illumination coefficients γ of the SH basis functions, as follows:

$$\mathcal{B}(\mathbf{n}, \mathbf{c} \mid \gamma) = \mathbf{c} \circ \sum_{b=1}^{B^2} \gamma_b \mathrm{Y}_b(\mathbf{n}) \ , \tag{7.2}$$

where $\mathrm{Y}_b(\mathbf{n}) \in \mathbb{R}$ is the b-th SH basis function evaluated on the surface orientation $\mathbf{n}$ and $\circ$ represents a point-wise multiplication. The irradiance is encoded using $B^2 = 9$ vector-valued SH illumination coefficients $\gamma = (\gamma_1^\top, \ldots, \gamma_{B^2}^\top)^\top$, where $\gamma_b = (\gamma_b^r, \gamma_b^g, \gamma_b^b)^\top \in \mathbb{R}^3$ denotes a vector that controls the irradiance separately for each color channel. This leads to $3 \cdot 9 = 27$ parameters in the proposed illumination model. A more detailed description of the image formation model and representation of the irradiance can be found in Section 2.2.2.

7.3.3 Coarse-scale Identity and Expression Model

The head is represented as a triangle mesh $\mathcal{M} = (\mathbf{V}, \mathbf{C}, \mathbf{G})$, where $\mathbf{V} = \{\mathbf{v}_n\}_{n=1}^N$ is the set of N vertices, $\mathbf{C} = \{\mathbf{c}_n\}_{n=1}^N$ is the set of per-vertex skin albedos, and $\mathbf{G} \subset \mathbf{V} \times \mathbf{V}$ denotes the mesh connectivity. In addition, we associate with each $\mathbf{v}_n$ a normal $\mathbf{n}_n$ which is computed based on its 1-ring neighborhood. The mesh's spatial embedding $\mathbf{V}$ and its per-vertex surface reflectance $\mathbf{C}$ is parametrized using the statistical head prior of Blanz and Vetter et al. [1999], which encodes the space of plausible human heads assuming a Gaussian distribution in the population. This linear head model is derived from 200 high-quality scans of Caucasian heads (100 males and 100 females) and compressed in a low-dimensional space using *principal component analysis* (PCA). Hence, vertex positions $\mathbf{v}_n = \mathcal{P}_n^s(\alpha)$ and skin reflectances $\mathbf{c}_n = \mathcal{P}_n^r(\beta)$, $\forall n$ can be parametrized as follows:

$$\text{Shape:} \mathcal{P}^s(\alpha) = \mathbf{a}_s + \mathbf{E}_s \Sigma_s \alpha \ , \tag{7.3}$$

$$\text{Reflectance:} \mathcal{P}^r(\beta) = \mathbf{a}_r + \mathbf{E}_r \Sigma_r \beta \ . \tag{7.4}$$

Here, $\mathbf{a}_s$, $\mathbf{a}_r \in \mathbb{R}^{3N}$ encode the per-vertex shape and reflectance of the average head, respectively. The shape and reflectance spaces are respectively spanned by the matrices $\mathbf{E}_s \in \mathbb{R}^{3N \times K_s}$ and $\mathbf{E}_r \in \mathbb{R}^{3N \times K_r}$, each containing the $K_s = K_r = 160$ first principal components of the shape and reflectance basis functions in their columns. Variations in shape and reflectance are controlled using the corresponding shape and reflectance parameters, $\alpha \in \mathbb{R}^{K_s}$ and $\beta \in \mathbb{R}^{K_r}$. The diagonal matrices $\Sigma_s = \text{diag}(\sigma_{\alpha_1}, \ldots, \sigma_{\alpha_{K_s}})$ and $\Sigma_r = \text{diag}(\sigma_{\beta_1}, \ldots, \sigma_{\beta_{K_r}})$ encode the standard deviations corresponding to the principal directions. Note that scaling the shape and reflectance bases by their standard deviations guarantees a similar range of variation for the control parameters. Normally, we search for identity parameters in the range $[-3\sigma_\bullet, +3\sigma_\bullet]$, $\bullet \in \{\alpha, \beta\}$, since this accounts for more than 99% of the variation and allows the model to discard unlikely head shapes and skin reflectances.

This linear shape model is extended to also cover facial expressions by adding $K_e = 75$ delta blendshapes (i. e., displacements from the rest pose) taken from a combination of the *Emily* model [Alexander et al. 2010] and the *FaceWarehouse* database [Cao et al. 2014b]:

$$\text{Expression: } \mathcal{P}^e(\alpha, \delta) = \mathcal{P}^s(\alpha) + \mathbf{E}_e \Sigma_e \delta \ , \tag{7.5}$$

where the matrix $\mathbf{E}_e \in \mathbb{R}^{3N \times K_e}$ contains the K_e delta blendshapes in its columns, $\delta \in [0,1]^{K_e}$ denote the expression weights and Σ_e is a diagonal matrix of empirically determined scale factors. Note that the delta blendshapes were transferred to the topology of the statistical shape model of Blanz and Vetter using deformation transfer [Sumner and Popović 2004]. It is also important to remark that the blendshapes in the Emily model are redundant (i. e., the rows of $\mathbf{E}_e$ are not linearly independent). As such, we employ a sparsity prior on δ, as described in Section 7.4.1.

7.3.4 Medium-scale Corrective Shapes

The coarse-scale model restricts the facial identity and expression to a $K_s = K_r = 160$ and $K_e = 75$ dimensional linear subspace, respectively. Variations falling outside of this low-dimensional subspace cannot readily be expressed with the model. Li et al. [2013b] and Bouaziz et al. [2013] showed that it is beneficial to leave this limited subspace to model characteristics in physiognomy and expression. In the spirit of [Bouaziz et al. 2013], we use *manifold harmonics* func-

tions [Vallet and Lévy 2008; Lévy and Zhang 2010] to parametrize a medium-scale 3D deformation field:

$$\text{Correctives: } \mathcal{P}^c(\tau) = \mathbf{E}_c\tau \ . \tag{7.6}$$

Here, $\mathbf{E}_c = [H_1 \otimes I_{3\times3}, \ldots, H_{K_c} \otimes I_{3\times3}] \in \mathbb{R}^{3N\times3K_c}$ contains three copies of the K_c linear *manifold harmonics* basis functions $H_k \in \mathbb{R}^N$ as columns and the parameters $\tau = [\tau_1^\top, \ldots, \tau_{K_c}^\top]^\top$ allow the control of the shape of the deformation field. Since a full 3D deformation field is required to control the corrective layer, each deformation coefficient $\tau_k \in \mathbb{R}^3$ is a vector itself. Note that the graph harmonics form a spectral basis that generalizes the *Fourier Transform* to the mesh domain. Here, $H_k, \forall k$ represent the $K_c = 80$ lowest-frequency eigenvectors of the Laplace Beltrami operator Δ_B computed on the estimated neutral shape $\mathcal{P}^s(\alpha)$ of the actor's face. We use *cotan*-weights to discretize Δ_B and obtain a symmetric positive semi-definite linear operator. The eigenvectors are efficiently computed using the band-by-band *shift invert spectral transform*, as suggested in [Lévy and Zhang 2010; Vallet and Lévy 2008]. Note that the lowest-frequency eigenvector H_1 has zero eigenvalue and therefore the first three columns of $\mathbf{E}_c$ will just represent a global 3D translation in $\mathcal{P}^c$. In view of this, H_1 was discarded from the spectral basis $\mathbf{E}_c$. Having estimated the correctives parameters τ, we can then apply the resulting 3D deformation field on vertex level, i. e., $\mathbf{v}_n + \mathcal{P}_n^c(\tau)$, where $\mathbf{v}_n = \mathcal{P}_n^e(\alpha, \delta), \forall n$ is a vertex of the coarse-scale model $\mathcal{M}^C$. Applying such deformation field then results in a medium-scale model $\mathcal{M}^M$.

Note that Bouaziz et al. [2013] infer correctives based on RGB-D data, while the proposed method robustly estimates them from RGB video alone (see Section 7.4.1). It is important to remark that the recent method of Ichim et al. [2015] does not estimate correctives from RGB video but modify the blendshapes themselves during tracking; however, they mention that capturing full correctives, as proposed here, will lead to better model personalization, but at the expense of a more involved optimization.

7.3.5 Fine-scale Detail Layer

Correctives are well suited to capture medium-scale detail variations among individuals, but lack the ability to represent static and transient fine-scale surface detail, such as wrinkles and folds. To alleviate this problem, we make use of an additional per-vertex displacement field to account for such effects. These fine-scale deformations are encoded in the gradient domain based on deformation gradients [Sumner and Popović 2004], which capture the non-translational surface deformation, including rotation, scale and shear.

Let $\hat{\mathbf{v}}^{(i)}$, respectively $\tilde{\mathbf{v}}^{(i)}$, $\forall i \in \{1,2,3\}$ be three 3D vertices of a triangle in the medium-scale $\mathcal{M}^M$ and fine-scale $\mathcal{M}^F$ mesh, and $\hat{\mathbf{n}}$, $\tilde{\mathbf{n}}$ their corresponding medium-scale and fine-scale surface normals. Then, the 3×3 affine deformation gradient $\mathbf{A}$ between the triangle faces of the two meshes is given as the solution of the following linear system:

$$\mathbf{A}\cdot(\hat{\mathbf{v}}^{(2)} - \hat{\mathbf{v}}^{(1)}, \hat{\mathbf{v}}^{(3)} - \hat{\mathbf{v}}^{(1)}, \hat{\mathbf{n}}) = (\tilde{\mathbf{v}}^{(2)} - \tilde{\mathbf{v}}^{(1)}, \tilde{\mathbf{v}}^{(3)} - \tilde{\mathbf{v}}^{(1)}, \tilde{\mathbf{n}}) \ . \tag{7.7}$$

Since rotation, scale and shear are inherently coupled in the per-face deformation gradients $\{\mathbf{A}_j\}_{j=1}^J$, where J is the number of triangles in the mesh, this representation does not allow for direct linear interpolation. We use polar decomposition [Higham 1986] to decompose the affine matrices $\mathbf{A}_j = \mathbf{Q}_j\mathbf{S}_j$ into their rotation $\mathbf{Q}_j$ and shear $\mathbf{S}_j$ components, and parametrize $\mathbf{Q}_j$ based on the matrix

exponential (3 parameters) [Alexa 2002]. From $\mathbf{S}_j$ we extract the scaling factors (3 parameters) and the skewing factors (3 parameters), which represent the scale and parallel distortion along the coordinate axes, respectively. In total, this leads to 9 parameters per triangle, each allowing for simple direct linear interpolation. These per-face representations are stacked in a feature vector $\mathbf{p} \in \mathbb{R}^{9J}$, which is used for parametrization and interpolation of surface detail of the fine-scale layer. As it will be shown in Chapter 8, such representation is very convenient for learning and generating fine-scale skin details of an actor's face rig.

7.4 Coarse- and Medium-scale Layer Reconstruction

For a given video $\mathcal{F} = (f_t)_{t=1}^{T}$ of T image frames f_t, the goal is to find the coarse- and medium-scale parameters of our personalized parametric 3D model that best explain the observed face in the scene. For the problem at hand, the recovery of the rigid head pose $(\mathbf{R}, \mathbf{t})$, the illumination γ, and the coarse (α, β, δ) and medium-scale parameters τ is expressed as an energy minimization problem, described as follows.

7.4.1 Energy Minimization

The model parameters $\mathcal{X} = (\mathbf{R}, \mathbf{t}, \alpha, \beta, \gamma, \delta, \tau)$ in $\mathrm{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^{K_s} \times \mathbb{R}^{K_r} \times \mathbb{R}^{3B^2} \times \mathbb{R}^{K_e} \times \mathbb{R}^{3K_c}$ are estimated based on an *analysis-by-synthesis* approach that maximizes the similarity between a synthetically generated image of the head and an input RGB frame f_t. This is formulated as a constrained multi-objective optimization problem:

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \Big[E_{data}(\mathcal{X}) + E_{prior}(\alpha, \beta, \gamma, \delta, \tau) \Big] \ , \tag{7.8}$$

$$\text{s.t.} : \mathbf{0} \leq \delta \leq \mathbf{1} \ . \tag{7.9}$$

The data objective E_{data} measures the photo-consistency and facial feature alignment of the synthetically generated image w. r. t. the input frame f_t. E_{prior} is a statistical prior that takes into account the likelihood of the identity and expression estimate. A *box*-constraint is imposed on the expression parameters δ to keep them in the range $[0, 1]$. To make the optimization more tractable, the hard *box*-constraint on the expression parameters in Equation 7.9 is relaxed and modeled as a soft-constraint E_{bound} directly in the reconstruction energy E_{total} (see Equation 7.10 below). This leads to the following unconstrained non-linear optimization problem:

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \underbrace{\Big[E_{data}(\mathcal{X}) + E_{prior}(\alpha, \beta, \gamma, \delta, \tau) + E_{bound}(\delta) \Big]}_{E_{total}(\mathcal{X})} \ . \tag{7.10}$$

Data Objective The data term measures how well the personalized 3D model explains the input frame f_t. To this end, we consider a photo-consistency measure E_{photo} as well as the alignment to salient facial features points $E_{feature}$:

$$E_{data}(\mathcal{X}) = w_f E_{feature}(\mathcal{X}) + w_p E_{photo}(\mathcal{X}) \ . \tag{7.11}$$

The weights w_f and w_p control the relative importance of these two objectives. Photo-consistency is measured on a per-vertex level. At vertex $\hat{\mathbf{v}}_n = \mathcal{P}_n^e(\alpha,\delta) + \mathcal{P}_n^c(\tau)$, with associated reflectance $\mathbf{c}_n = \mathcal{P}_n^r(\beta)$ and normal $\hat{\mathbf{n}}_n$ that depends on the same parameters, it compares the surface color $\mathcal{B}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)$ synthesized according to the irradiance model in Equation 7.2 with the actual color $f_t[\hat{\Pi}\mathcal{C}(\hat{\mathbf{v}}_n)]$ in the input image. The corresponding energy reads as follows:

$$E_{photo}(\mathcal{X}) = \sum_{n=1}^{N} \| f_t[\hat{\Pi}\mathcal{C}(\hat{\mathbf{v}}_n)] - \mathcal{B}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma) \|_2^2 \quad . \tag{7.12}$$

In addition, the alignment of salient facial features is also taken into account. To this end, we measure the distance between image projections $\{\hat{\Pi}\mathcal{C}(\hat{\mathbf{v}}_{n_\ell})\}_{\ell=1}^{L}$ of a selection of $L = 66$ feature vertices on the model and their corresponding distinct detected facial landmarks $\mathbf{Y} = \{\mathbf{y}_\ell\}_{\ell=1}^{L}$ in the input image:

$$E_{feature}(\mathcal{X}) = \sum_{\ell=1}^{L} \| \hat{\Pi}\mathcal{C}(\hat{\mathbf{v}}_{n_\ell}) - \mathbf{y}_\ell \|_2^2 \quad . \tag{7.13}$$

Note that the 2D facial features are tracked with an off-the-shelf algorithm [Saragih et al. 2011a] and their landmark trajectories are improved by utilizing optical flow between automatically selected key-frames, as described in Section 4.3. To select the 3D feature points $\{\hat{\mathbf{v}}_{n_\ell}\}$ on the model, we automate and extend the strategy proposed in Section 5.4.2. In a pre-processing step, $K_e = 75$ different facial expressions of the average person are synthesized by activating one expression weight δ_k at a time and frontal views of the resulting meshes are rendered under a fixed user-defined illumination. Afterward, the off-the-shelf face tracker is employed to detect the 2D landmarks in the synthetically generated images. The 2D landmarks are then back-projected to the nearest vertices on the 3D model, discarding those that fall outside of the face region or inside the mouth cavity. Finally, the 3D positions corresponding to the same landmark are averaged and assigned to the nearest valid vertex of the model.

Prior Objective 3D reconstruction from monocular RGB input is an ill-posed problem due to its inherent depth ambiguity. As a result, many spatial configurations of mesh vertices lead to a similar projection in the camera. This issue is tackled by incorporating suitable priors E_{prior} into the energy of Equation 7.10. This allows us to disambiguate reasonable from unreasonable configurations and steer the optimization into the right direction. To this end, we employ two probabilistic shape priors (E_{prob_1}, E_{prob_2}) and a sparsity prior E_{sparse} on the expression coefficients:

$$E_{prior}(\alpha,\beta,\gamma,\delta,\tau) = E_{prob_1}(\alpha,\beta,\gamma) + E_{prob_2}(\tau) + E_{sparse}(\delta) \quad . \tag{7.14}$$

The probability of a certain scene configuration is accounted for by assuming multiple Gaussian distributions over the parameters:

$$E_{prob1}(\alpha,\beta,\gamma) = w_s \sum_{k=1}^{K_s} \left(\frac{\alpha_k}{\sigma_{\alpha_k}}\right)^2 + w_r \sum_{k=1}^{K_r} \left(\frac{\beta_k}{\sigma_{\beta_k}}\right)^2 + w_l \sum_{b=1}^{B^2} \left\| \frac{\gamma_b}{\sigma_{\gamma_b}} \right\|_2^2 \quad , \tag{7.15}$$

with the division in the last term being component-wise. Here, w_s, w_r and w_l weigh the different objectives. As in [Blanz and Vetter 1999; Zollhöfer et al. 2014], the shape weights α and reflectance

coefficients β are restricted to stay statistically close to the mean using L_2-regularization. Since the standard deviations of the lighting coefficients γ are unknown, *Tikhonov*-regularization constraints [Hoerl and Kennard 2000] are imposed instead by setting $\sigma_{\gamma_b} = [1,1,1]^\top$.

In addition, the medium-scale shape correctives parameters are regularized based on their standard deviations (squared eigenvalues of the *manifold harmonics* functions H_k, $\forall k$ described in Section 7.3.4) and temporal smoothness w. r. t. the corresponding result of the previous frame τ^{prev} is further enforced, as follows:

$$E_{prob2}(\tau) = w_z \sum_{k=1}^{K_c} \left\| \frac{\tau_k}{\sigma_{\tau_k}} \right\|_2^2 + w_t \|\tau - \tau^{prev}\|_2^2 \ , \tag{7.16}$$

with component-wise divisions in the first term. Here, w_z and w_t are the weights controlling the importance of the different objectives.

Following [Bouaziz et al. 2013], we also impose L_1-regularization on the expression weights δ to enforce sparsity. This avoids potential blendshape compensation artifacts due to the inherent redundancy in the expression basis:

$$E_{sparse}(\delta) = w_d \sum_{k=1}^{K_e} |\delta_k| \ . \tag{7.17}$$

Boundary Constraint The blendshape parameters are restricted to a reasonable range ($\delta_k \in [0,1]$) by adding a soft *box*-constraint with a weight of w_b to the energy:

$$E_{bound}(\delta) = w_b \sum_{k=1}^{K_e} \phi(\delta_k) \ . \tag{7.18}$$

The function ϕ adds a penalty to the energy if and only if its parameter leaves the trusted region:

$$\phi(x) = \begin{cases} x^2 & \text{if } x < 0, \\ 0 & \text{if } 0 \le x \le 1, \\ (x-1)^2 & \text{if } x > 1. \end{cases} \tag{7.19}$$

We use a symmetric quadratic penalizer outside of the trusted region to tightly enforce the bounds of this constraint.

7.5 Fine-scale Layer Reconstruction

Given the medium-scale result $\mathcal{M}^M$ at each frame f_t, $\forall t$ of the previous optimization, fine-scale static and transient surface details (i. e., wrinkles and folds) are recovered from shading cues in the input RGB images by adapting the shading-based refinement approach under unknown lighting and albedo presented in Section 5.6.

We compute shading-based refinement on a per-vertex level, yielding a high-quality refined mesh $\mathcal{M}^F$. The previously estimated reflectance and illumination is utilized as initialization. A refinement optimization then adapts the mesh's vertex positions via inverse rendering optimization, such

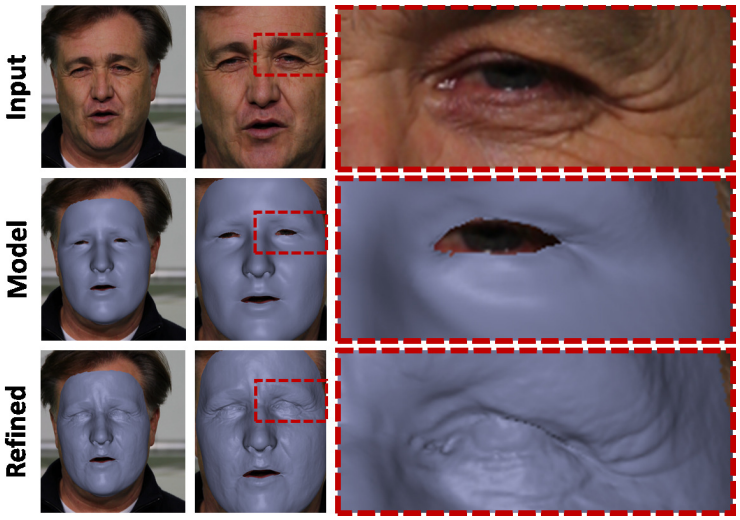


Figure 7.4: Fine-scale layer reconstruction. Shading cues in the input image (top) are exploited to augment the medium-scale model (middle) with fine-scale static and transient surface detail (bottom), thereby creating a fine-scale layer of details.

that the synthesized shading gradients match the gradients of the illumination in the corresponding input RGB image as well as possible. To further regularize this ill-posed problem, spatial and temporal detail smoothness is enforced as a soft constraint [Garrido et al. 2013; Valgaerts et al. 2012b]. The final vertex normals are computed by averaging over a temporal window of size 5 for stability [Nehab et al. 2005]. As mentioned in Section 7.3.5, the deformation field between the medium-scale result $\mathcal{M}^M$ and the refined high-quality geometry $\mathcal{M}^F$ is parametrized using the proposed deformation gradient-based feature vector representation $\mathbf{p}$. Compared to $\mathcal{M}^M$, the resulting high-quality reconstructions exhibit a considerable amount of fine-scale surface detail, as shown in Figure 7.4.

7.6 Multi-step Optimization Strategy

Given the input video $\mathcal{F} = \{f_t\}_{t=1}^{T}$, we find the best parameters $\mathcal{X}$ by minimizing the non-linear objective $E_{total}(\mathcal{X})$ of Equation 7.10 based on a multi-step optimization strategy which consists of several Levenberg-Marquardt [Levenberg 1944; Marquardt 1963; Moré 1978] optimization stages. Note that our Levenberg-Marquardt solver employs analytical partial derivatives to compute the Jacobian matrix, which is used to iteratively update the parameters $\mathcal{X}$. The partial derivatives of all the terms of $E_{total}(\mathcal{X})$ can be found in Appendix A (see Section A.2).

Algorithm 1 summarizes all the individual optimization steps to obtain the parameters of the multi-layer parametric 3D face model. In a pre-processing step, the rigid head pose ($\mathbf{R}$ and $\mathbf{t}$) is initialized using the POSIT algorithm [David et al. 2004] on the detected facial landmarks, and (α, δ) are initialized by solving Equation 7.13 with the parametric priors $E_{prob1}(\alpha)$, $E_{sparse}(\delta)$, and $E_{bound}(\delta)$, i. e., we optimize for (α, δ) using only the facial feature point subspace. The other parameters (β, γ, τ) are initially set to zero.

After the initialization step, the first $T_{first} \approx 100$ frames of the sequence are utilized to reconstruct a coarse-scale estimate of the actor's person-specific identity (α, β), as well as the illumination γ in the scene. This step does not consider the corrective parameters τ; therefore, the corresponding

Algorithm 1 Multi-Step Optimization Strategy

```
(R, t, α, β, γ, δ, τ) ← Initialize();

for (the first T_first frames f_t) do                          ▷ Identity Estimation
    while (not converged) do
        (R, t)       ← Estimate_Head_Pose();
        (α, β, γ)    ← Estimate_Identity_And_Illumination();
        (δ)          ← Estimate_Expression();
    end while
end for

(C_p) ← Build_Person_Specific_Albedo_Map();

lighting_opt = get_lighting_option();
for every frame f_t ∈ F do
    while (not converged) do                                   ▷ Coarse-Scale
        (R, t) ← Estimate_Head_Pose();
        (δ)    ← Estimate_Expression();
        if lighting_opt == per_frame then
            (γ) ← Estimate_Illumination();
        end if
    end while
    while (not converged) do                                   ▷ Medium-Scale
        (R, t) ← Estimate_Head_Pose();
        (τ)    ← Estimate_Correctives();
        if lighting_opt == per_frame then
            (γ) ← Estimate_Illumination();
        end if
    end while
    (p) ← Compute_Detail_Layer();                              ▷ Fine-Scale
end for
```

terms are removed from the energy. The resulting per-frame estimates of the actor's identity are combined using a floating average.

Before tracking the complete sequence in the next stage, an actor-specific skin reflectance map $\mathbf{C}_p$ is generated which replaces the per-vertex reflectance estimates from the parametric actor model. To this end, we follow a similar strategy described in Section 6.7.1 and compute per-pixel albedo values by dividing through the lighting term (sum on the right hand side of Equation 7.2) on a subset of 10 frames. The resulting albedo values are averaged in the final map $\mathbf{C}_p$ using the aligned model. This refined appearance step drastically improves the subsequent tracking performance, since the generated reflectance map resembles the actor's appearance much better (i. e., facial hair and fine-scale skin detail are explicitly accounted for, see also [Zollhöfer et al. 2014]). Figure 7.5 shows that the personalized albedo map rendered under the estimated scene lighting captures more fine-scale albedo variations than the low-dimensional parametric model $\mathcal{P}^r(\beta)$, which in turn helps improve the tracking (see for instance the details around the eyes and the mouth shape of the actor).

After estimating the identity and computing the personalized albedo map, the identity parameters

Figure 7.5: Parametric reflectance model vs. personalized texture map. In contrast to the low-dimensional parametric face model, the automatically computed personalized texture map captures fine-scale albedo variations and contributes to a better fitting.

α are kept fixed and the complete sequence is tracked again, starting from the first frame. For each frame f_t, we first re-estimate the head pose $(\mathbf{R},\mathbf{t})$ and compute the best fitting blendshape coefficients δ. The coarse-scale shape estimate and the head pose are then improved by optimizing for the best corrective parameters τ, as well as $\mathbf{R}$ and $\mathbf{t}$, based on the full reconstruction energy (see Equation 7.8). Note that in this improvement step the blendshape coefficients δ stay fixed. It is important to remark that the scene lighting can be re-estimated at each frame f_t when estimating the coarse- and medium-scale shape, if desired. Even though the fitting may improve with a refined scene illumination, a high-frequency color flicker on the synthesized face was detected in the temporal domain and therefore the lighting was kept constant for the entire sequence. This is a sometimes incorrect but fair assumption that works in most scenes and that was also used in other related approaches, e. g., in [Shi et al. 2014]. The final step reconstructs a parametric fine-scale detail layer $\mathbf{p}$ based on shading-based shape refinement by exploiting shading cues in the input RGB frame.

7.7 Experiments

In this section, we present a qualitative and quantitative evaluation, and perform a thorough comparison w. r. t. the state-of-the-art in monocular face reconstruction. The robustness of the approach presented in this chapter is demonstrated for a wide range of scenarios, from controlled studio setups to uncontrolled legacy video footage. In total, the proposed approach was evaluated on 9 test sequences: Three indoor sequences captured in a controlled setup (SUBJECT1, SUBJECT2, SUBJECT3), two outdoor sequences (SUBJECT4, SUBJECT5) and four legacy videos (ARNOLD YOUNG, ARNOLD OLD, OBAMA, BRYAN) freely available on the Internet and downloaded from YouTube. Please refer to Appendix A (see Section A.1) to find more information about the sequences. The reconstructed personalized 3D face models consist of $N = 200$k vertices and $J = 400$k triangle faces.

Most of the results and comparisons shown below are viewed best as video. Hence, the reader is strongly advised to watch the supplemental videos at the project website[2].

[2] http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig/

Parameters and Runtimes The proposed facial performance capture approach relies on weights that specify the relative importance of the different objectives. During the performed tests, it turned out that our approach is insensitive to the specific choice of parameters in the different sequences. The following weights were then fixed and used in all the experiments: $w_f = 0.5$, $w_p = 1$, $w_s = 0.01$, $w_r = 1$, $w_l = 0.1$, $w_z = 40$, $w_t = 4$, $w_d = 100$ and $w_b = 10^9$.

The approach described above was implemented on the CPU using simple parallelization routines in OpenMP. Overall, this implementation takes several hours to process a sequence of 1k frames when executed on an Intel Core i7-3770 CPU (3.4 GHz). For each processed frame, the proposed method requires 30 ms for facial landmark extraction, 1.5 s for landmark refinement, 40 s for identity fitting (only run for the first 100 frames), 15 s for coarse-layer tracking, 9 s for medium-layer correctives and 110 s for fine-scale shape refinement. Note that the last step was not optimized on the CPU and runs on a single core.

By harnessing the data parallel processing power of modern GPUs, we have discovered in a recent project (not part of this thesis) that a drastic reduction of the computation time is possible. This has also been corroborated recently by different approaches that solve similar non-linear optimization problems [Thies et al. 2015; Wu et al. 2014; Zollhöfer et al. 2014].

7.7.1 Qualitative and Quantitative Results

The proposed multilayer model-based approach estimates the actor's facial identity and tracks his/her facial expressions, where tracking progresses in a coarse-to-fine manner on the three layers: Coarse-scale shape, medium-scale correctives and fine-scale wrinkle-level detail. Figure 7.6 shows the output tracking results of the personalized 3D face reconstruction of OBAMA and ARNOLD OLD for the three different layers of personalization. Note that the medium- and fine-scale layers do not only lead to more realistic results in terms of high-frequency detail, but also deliver tracking results of superior accuracy (please refer to the supplementary video available at the project website to see a more interactive comparison between the reconstructed layers).

In addition, we quantitatively evaluate the geometric reconstruction accuracy obtained by the proposed approach on the sequence of SUBJECT1. For this indoor sequence, two views are available and high-quality ground truth 3D meshes have been generated using the binocular facial performance capture approach of Valgaerts et al. [2012b]. Note that only one of the two views was employed to reconstruct the personalized parametric 3D model of the actor. Figure 7.7 shows a comparison between our monocular method and the binocular stereo approach of Valgaerts et al. [2012b] for a reconstruction of SUBJECT1 at neutral pose; therefore, it shows how well the proposed method can reconstruct the identity of the actor. As illustrated in the heatmap overlay, the geometric error (computed as the per-vertex Euclidean distance of the aligned reconstructions) is quite small (1.8 mm on average) and errors mainly appear in the nose region, stemming from depth inaccuracies.

7.7.2 Validations: Comparison to Performance Capture Approaches

In this section, we compare the reconstruction quality of the proposed monocular approach to existing multiview and monocular facial performance capture methods proposed in the literature (please also refer to the second supplementary video available at the project website to examine the comparisons in more detail). It is important to remark that none of these methods capture a fully-parametrized 3D model at different personalization layers: Coarse, medium and fine. As it will be seen in Chapter 8, such a convenient parametrization will enable us to create a fully-controllable

Figure 7.6: Facial performance capture results generated on OBAMA (left) and ARNOLD OLD (right) sequence. Given a monocular video of a person as input (first row), the proposed approach achieves high-quality reconstructions (both identity and facial motion) on multiple parametrized layers: Coarse-scale shape and motion (second row), medium-scale correctives (third row) and fine-scale wrinkle-level surface detail (forth row).

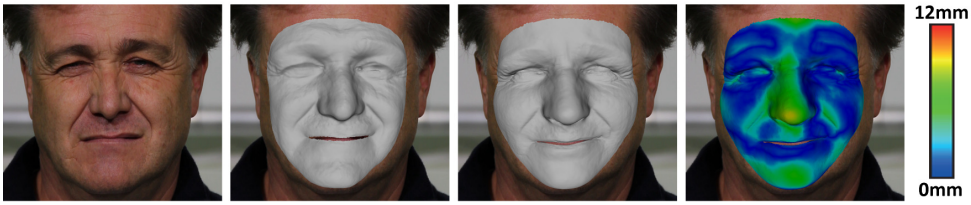


Figure 7.7: Geometric accuracy - SUBJECT1. *Left to right:* Input (neutral pose), binocular stereo reconstruction, our monocular reconstruction, geometric error represented as a heatmap overlay. The proposed multilayer approach obtains a detailed 3D model of similar quality (1.8 mm mean error) compared to the binocular stereo reconstruction of Valgaerts et al. [2012b].

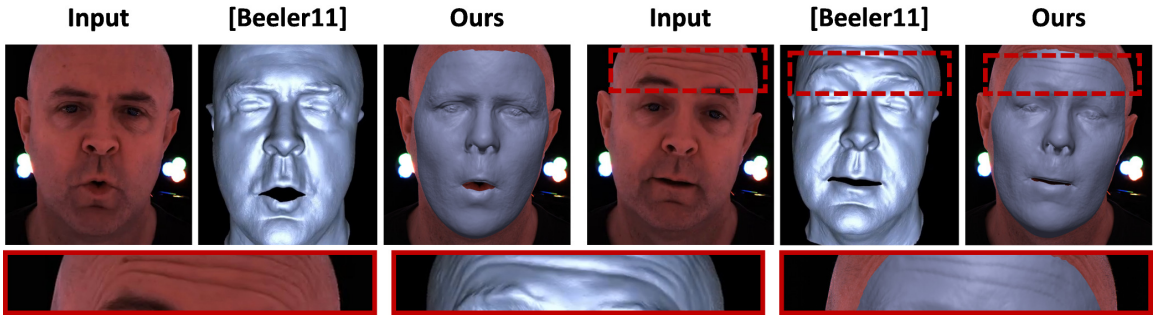


Figure 7.8: State-of-the-art comparison to the multiview in-studio approach by [Beeler et al. 2011] - SUBJECT3. The proposed multilayer method, which reconstructs detailed geometry from a single video under general lighting, comes close in reconstruction quality to that of Beeler et al.'s method which requires a professional setup with 6 high-quality cameras.

generative 3D model for person-specific expression and wrinkle synthesis in a flexible way, which in turn will allow us to generate realistic personalized animations and perform complex video editing tasks.

Comparison to Beeler et al. 2011 Figure 7.8 shows a comparison with the high-quality off-line performance capture method of Beeler et al. [2011]. This method requires a controlled setup with 6 high-quality cameras and controlled in-studio lighting to perform a variant of multiview stereo in combination with a mesoscopic detail augmentation step. In contrast, the proposed method takes as input a single monocular video under general lighting and is capable of achieving a reconstruction quality that comes close to their approach.

Comparison to Cao et al. 2014a The state-of-the-art monocular performance capture approach of Cao et al. [2014a] is able to reconstruct the actor's identity and motion at a coarse-scale. While reconstructions can be obtained at video rate, they lack fine-scale surface detail and do not capture actor-specific idiosyncrasies in identity and motion, as it can be seen Figure 7.9. In contrast, the proposed off-line approach reconstructs person-specific surface detail at a medium- and fine-scale level of personalization, leading to high-quality reconstructions that fit the input face in the video more closely.

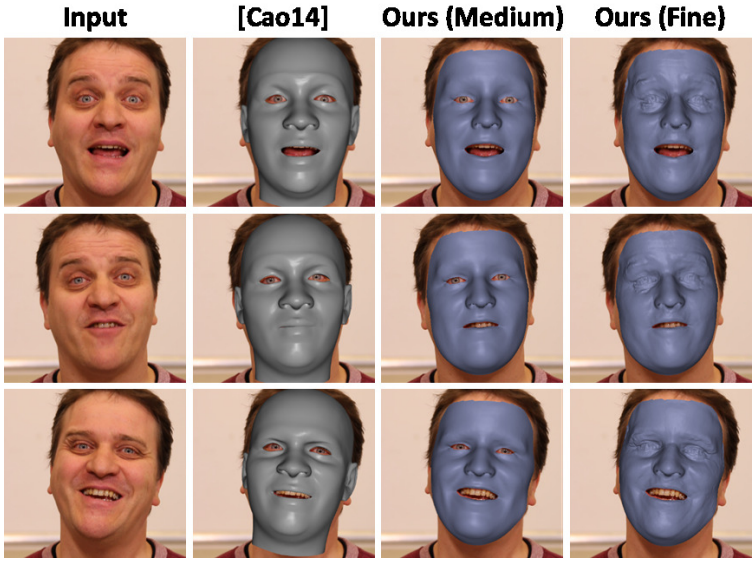


Figure 7.9: State-of-the-art comparison to [Cao et al. 2014a] - SUBJECT2. *From left to right:* Monocular input, result obtained by the approach of Cao et al., our medium-scale result and our final fine-scale reconstruction. Note that the medium-scale result matches the actor's facial geometry better and the fine-scale reconstruction adds even more realism.

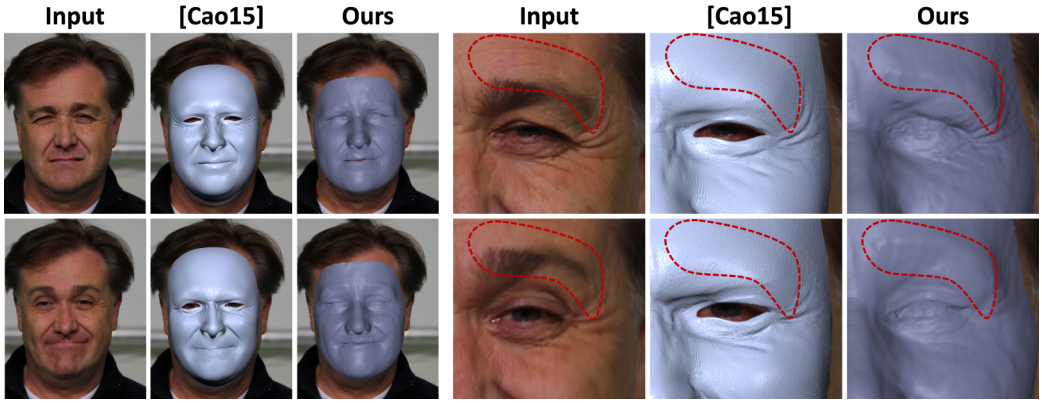


Figure 7.10: State-of-the-art comparison to [Cao et al. 2015] - SUBJECT1. While the regression-based approach of Cao et al. infers some of the actor's fine-scale details, it produces less accurate results if poses and identities are far from the training set. In particular, note the overall less accurate reconstruction of identity (left), as well as the imprecise reconstruction of some wrinkles and the shape of the eyebrow (right). In contrast, our reconstruction-based approach delivers results closer to the real input video. Please note that fine-scale pores in the results of Cao et al. are merely hallucinated, as they are part of the model learned from high-quality face scans.

Comparison to Cao et al. 2015 In [Cao et al. 2015], an extension to [Cao et al. 2014a] that additionally regresses a wrinkle-level displacement map has been proposed. This approach learns the correlation between image patches and surface detail from a database of 3D scans. While this augments the coarse-scale reconstruction with detail, the inferred geometry is not metrically correct. Thanks to the medium-scale corrective layer, our personalized multilayer face model overlays with the input better, even if the fine-scale detail is ignored for a moment (see Figure 7.10). Furthermore, the inverse rendering approach presented here allows us to obtain detail reconstructions that match

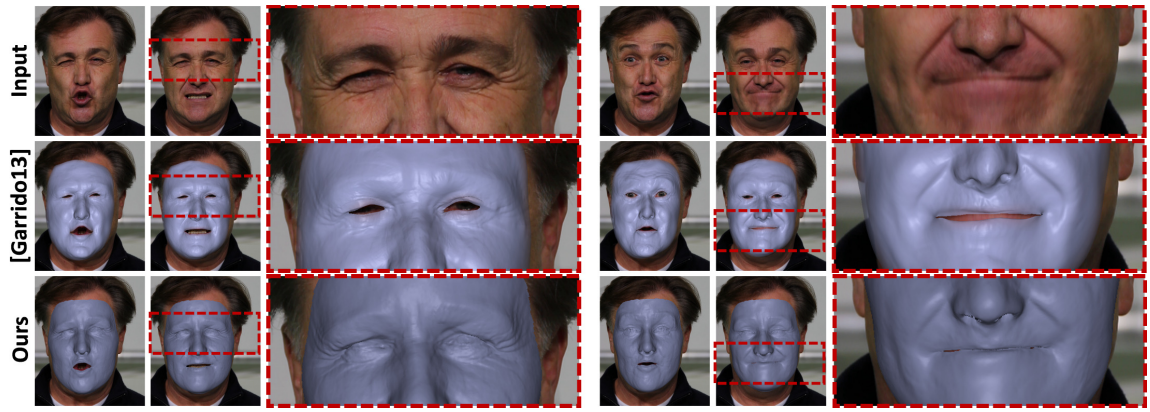


Figure 7.11: State-of-the-art comparison to method presented in Chapter 5 - SUBJECT1. Compared to the method proposed for semi-constrained setups, our multilayer-based reconstructions better match the actor's static and transient small-scale surface details. Note that the previous method requires a high-quality laser scan of the actor as input, making it unsuitable for legacy video footage.

the true detail in the image closer than the regression result, which can only approximate as close as possible (see especially the shape of the eyebrows in the figure).

Note that the fine-scale pores that appear on the meshes from Cao et al. [2015] are not reconstructed, but only part of the high-quality template model used for learning their representation. Furthermore, as their detail regression approach is based on cues in the input image, it cannot generate a detail layer for an arbitrary novel expression specified by user-defined blendshape weights, which is normally required by animation artists as a de facto standard to create new realistic facial animations. The proposed multilayer parametric method could in principle generate such details if a proper mapping between the coarse and the other layers is provided. Chapter 8 presents such a method that leverages the inherent semantics of the blendshape weights to learn person-specific characteristics (including details) from the personalization layers captured by the current approach, making it suitable for the scenario described above.

Comparison to Chapter 5 As shown in Figure 7.11 and Figure 7.12, the proposed multilayer approach is able to obtain similar or even higher quality reconstructions than those of the monocular state-of-the-art facial performance capture method presented in Chapter 5. Furthermore, strong out-of-plane head rotations can be handled much better by the current approach. Even though both methods are able to track facial expressions and reconstruct fine-scale surface detail, the method presented in Chapter 5 requires calibrated cameras and heavily relies on a static high-quality 3D scan of the actor as prior which is not always available. Therefore, unlike the proposed method presented in this chapter, the previous approach is not applicable to cases where only legacy footage is available. Also note that the current multilayer parametric representation is able to decouple medium-scale corrective shapes from rigid pose, which turned out to be quite convenient when learning a face rig for person-specific idiosyncrasy and detail generation, as it will be seen in Chapter 8.

Comparison to Shi et al. 2014 Finally, we compare to the high-quality monocular approach of Shi et al. [2014]. Their method employs a multilinear model for face reconstruction and can also be applied to legacy footage, as shown in Figure 7.13. The proposed multilayer approach is able to attain higher-quality reconstructions on the coarse as well as on the fine-scale due to the use

Figure 7.12: State-of-the-art comparison to the approach by [Shi et al. 2014] and [Garrido et al. 2013] - SUBJECT4. Our monocular approach obtains better reconstruction quality than that of Shi et al.'s and Garrido et al.'s method. Note the better tracking on the coarse geometry as well as on the fine-scale detail layer.

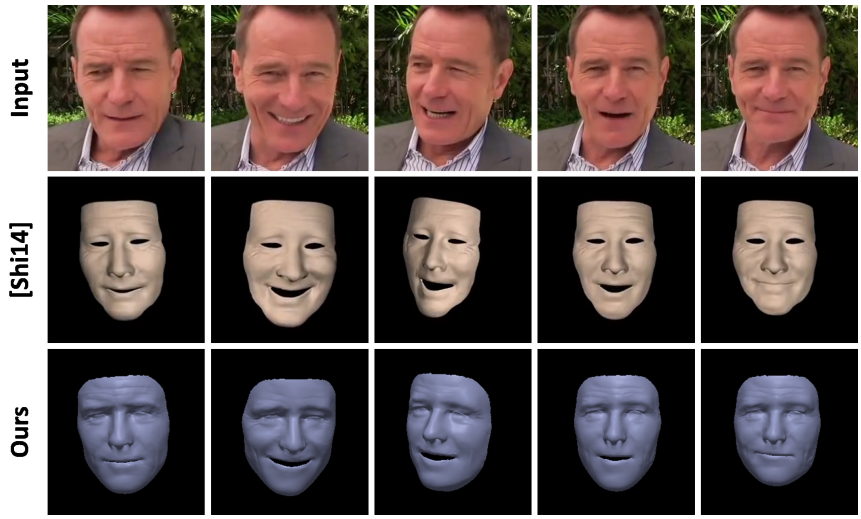


Figure 7.13: State-of-the-art comparison to the approach by [Shi et al. 2014] - BRYAN. Our approach obtains a closer fit than Shi et al.'s method. Note that the amount of fine-scale surface detail captured by the multilayer model is much higher.

of dense correspondences to jointly optimize for identity and expression. Moreover, our approach obtains a better model personalization and accurately tracks facial motion by employing medium-scale corrective shapes. On the other hand, Shi et al. mainly resort to sparse correspondences to estimate large-scale deformations, which are then slightly improved using normal maps estimated in their shade-from-shading framework. This leads to a less accurate head pose, as well as less accurate coarse- and fine-scale surface reconstructions, as illustrated in Figure 7.12. It is important to

remark that in the method of Shi et al. medium- and fine-scale deformations are mixed together and not parametrized, making it unsuitable for complex video editing tasks other than texture modification. Our fully-parametrized approach, on the other hand, could potentially be used for such tasks, though in a not very intuitive way. In Chapter 8 we learn a correlation model for person-specific correctives and wrinkles, thus allowing us to automatically adapt the captured detailed layers to match person-specific idiosyncrasies in accordance to the activated expression, which is the foundation for realistic video editing.

7.8 Discussion and Limitations

In this chapter, we have demonstrated that 3D facial geometry can accurately be reconstructed at multiple parametric face layers in unconstrained setups, such as legacy footage downloaded from YouTube. While these results are compelling and compete with (or sometimes overcome) those of state-of-the-art monocular methods even in challenging cases (e. g., fast head motion, expressive faces, and out-of-plane head rotations), the proposed method still has some limitations: Since our reconstruction approach is based on temporal frame-to-frame coherence, videos that exhibit lots of cuts are hard to handle automatically, requiring re-initialization of the parameters. Reconstructing multiple actors from a single video also requires an extra face detection and recognition component to keep the approach automatic. Even though the reconstructed layers are fully parametrized and could potentially be used for facial animation and editing tasks, the medium- and fine-scale layers do not provide any semantic parametrization nor are they correlated to blendshapes to allow an intuitive control of the personalized 3D model. It would be much more desirable to indirectly manipulate the layers through blendshape weights with which animation artists are more familiar. Chapter 8 addresses this problem by learning a correlation model that couples standard expressions in the blendshape weight domain with person-specific correctives and details, thereby resulting in a personalized face rig that synthesizes new detailed expressions by simple modifying blendshape controllers.

As in Chapters 5–6, the proposed approach also assumes *Lambertian* reflectance. Although this is a fairly common assumption which has been established in the literature [Ichim et al. 2015; Shi et al. 2014; Suwajanakorn et al. 2014; Valgaerts et al. 2012b], it introduces artifacts in the presence of specular highlights, as shown in Figure 7.14 (a). Consequently, subsurface scattering effects are not modeled; instead, the scene's light transport is parametrized using a low-dimensional SH representation which assumes smooth distant illumination and no shadows. As such, extreme lighting (e. g., directional spotlights) and cast shadows lead to artifacts.

Mild occlusions on the face, such as hair can be handled by the current approach, but occluding objects may be wrongly captured as facial features, both in the texture map and in the reconstructed layers (especially the fine layer). Strong occlusions, such as a dense beard, pose a problem to both the 2D face tracker and the identity reconstruction (non-skin reflectance and occluding objects are not explained by our statistical prior). The optimization of the medium layer relies on global-support corrective functions to correct tracking residuals and assumes that all facial features contribute equally. Thus, face tracking is challenged by fast and complex local facial deformations, especially in the mouth region, as shown in Figure 7.14 (b). Additional constraints and a semantic basis for local corrections may alleviate this problem, but reconstructing accurate mouth/lip shape is still challenging due to depth ambiguities and disocclusions of the lips. In Chapter 9, we address the problem of accurate lip tracking from monocular video and propose a data-driven approach that learns a robust lip shape regressor from high-quality multiview reconstructions to enhance the lip

(a) (b)

Figure 7.14: Limitations. Reconstruction artifacts due to (a) specular highlights on the face and (b) the lack of a local-support corrective basis and constraints to handle mouth deformations.

shape and motion of tracked facial performances.

Finally, we share the limitation of related work that no detailed mouth interior or eye/eyelid model can be reconstructed from video alone. Therefore, eye geometry or blendshapes for blinking were not modeled; the latter was reconstructed as shape detail in the shading-based refinement step.

7.9 Summary

This chapter has presented a new fully automatic model-based approach that can accurately capture facial shape and expressions at multiple personalization layers (from coarse-scale shape up to a layer that accounts for wrinkles and folds) in completely unconstrained videos where all scene elements are unknown. The heart of this approach is a new multilayer parametric 3D face prior that jointly encodes plausible appearance and shape changes in a low-dimensional space to model the image formation process. A novel unified fitting approach that leverages both color cues and sparse 2D landmarks is used to accurately reconstruct the overall shape and appearance at the coarse and medium layer. A fine-scale wrinkle model reconstructed over the face surface constitutes the final most detailed layer. Overall, all the three layers are optimized in an inverse rendering framework, making the method elegant and very robust. Qualitative and quantitative results have demonstrated the high fidelity of the reconstructed parametric models for several actors from different sources of video, including indoor, outdoor and YouTube footage.

The algorithmic improvements presented in this chapter considerably advance the state of the art in unconstrained facial performance capture from monocular video. As it will be shown in Chapter 8, the reconstructed multilayer parametric 3D models will constitute the basis for learning a fully-controllable personalized face rig that can be used by digital artists for achieving complex facial animation and video editing tasks.

Chapter 8

Beyond Face Capture: Face Rig Creation, Animation and Editing

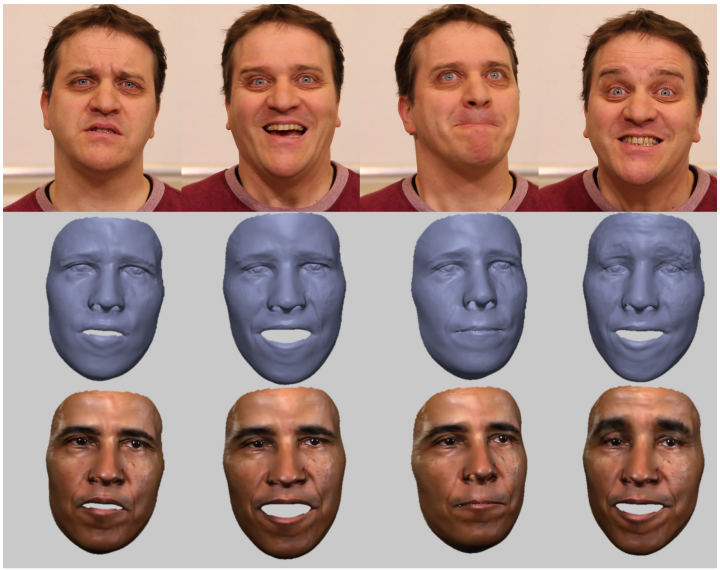


Figure 8.1: Fully personalized, detailed 3D face rig of the US president Barack Obama (bottom), reconstructed from the sequence shown in Figure 7.1. By transferring the estimated blendshape parameters from the input video (top), the facial rig can be used, for instance, for reenactment. Note that the rig preserves the idiosyncrasies of the president and *not* of the input face.

So far monocular face capture approaches can reconstruct detailed dynamic 3D face surfaces, but no method is able to create a face rig (i. e., a controllable model for person-specific expression and detail generation) from arbitrary monocular data. Such a rig would be beneficial for animation artists when no information about the actor is available other than a simple monocular video. This chapter presents a novel regression-based approach for the automatic creation of detailed, personalized 3D face rigs from arbitrary monocular performance capture data (Chapter 7) that can be conveniently driven by intuitive blendshape weights to perform video editing and animation tasks (see Figure 8.1). The method and results presented in this chapter are based on [Garrido et al. 2016a].

8.1 Introduction

The creation of believable face animations for virtual actors in movies and games, or for avatars in virtual reality or teleconferencing scenarios is a challenging task. Since our expert eye is attuned to detecting small inaccuracies in face appearance and motion, animation artists spend tremendous effort to model and animate high-quality facial animation rigs, especially in movies. A common practice for an artist is to design a face animation rig with custom-made control parameters that steer facial expression, face shape, and possibly face appearance and soft tissue deformation (see Section 2.1.2 for further details). The de facto standard to parametrize expression control is a blendshape model that linearly combines a set of basis expressions [Lewis et al. 2014]. Professional rigs are normally derived from detailed laser scans and often feature hundreds of control parameters (process that may take weeks of work). The resulting face rig is often animated from face mocap data, a step requiring frequent manual intervention.

To simplify this complex animation pipeline, researchers have developed different methods to automate some of its steps. For instance, some algorithms employ dense camera and lighting arrays to reconstruct dynamic face geometry and face appearance (Section 3.1.1), while other approaches extract components of face rigs from densely captured animation data, such as blendshape components [Neumann et al. 2013; Joshi et al. 2003]. Despite its practical relevance, automatic rig creation has received much less attention in the field. Meanwhile, performance capture methods were further extended to work with only a single RGB camera (Section 3.1.3) or a single RGB-D sensor that integrates both color and depth information (Section 3.1.2). However, there is still no approach that fully-automatically reconstructs and animates a detailed personalized modifiable face rig, from only a single RGB video of an actor filmed under general conditions (Section 3.3). Up to now, state-of-the-art lightweight approaches, as well as the methods presented in previous chapters, reconstruct default blendshape models that cannot capture and parametrize the person-specific nuances in face shape (i. e., identity) and expressions, and therefore they lack personalization at a finer level of detail.

In this chapter, we propose the first automatic method that builds a fully personalized, controllable 3D face rig (see Figure 8.1), given the multilayer 3D reconstructions captured in unconstrained monocular video by the method presented in Chapter 7. This personalized rig is based on the three distinct layers reconstructed in Chapter 7 (i. e., coarse, medium and fine layer) and learned by coupling the coarse layer (represented by generic blendshape weights) to the medium- and fine-scale detail layer via a novel sparse learning regression approach. This assures a semantic control of the detailed layers in ways consistent with the deformations of the coarse facial expressions (i. e., blendshapes). Hence, new expressions, even unseen ones, with proper fine-scale detail can be created for the face rig by simply modifying blendshape controllers or activation curves (scenario that fits nicely into an animator's standard workflow). The proposed regression method creates detailed, personalized face rigs from arbitrary monocular performance capture data, e. g., played by our favorite vintage actor, for which it would be impossible to automatically capture a rig by any other means.

The proposed method improves over existing state-of-the-art approaches in several important ways. Some previous methods employ generic parametric expression and identity models for monocular facial performance capture [Cao et al. 2014a; Shi et al. 2014], but they neither parametrize nor learn a generative model for person-specific expressions and fine-scale details observed in video. Generative models of face wrinkle formation have been learned from high-quality expressions (out of a vast set of examples) captured with a dense sensor array [Bermano et al. 2014; Cao et al. 2015] or

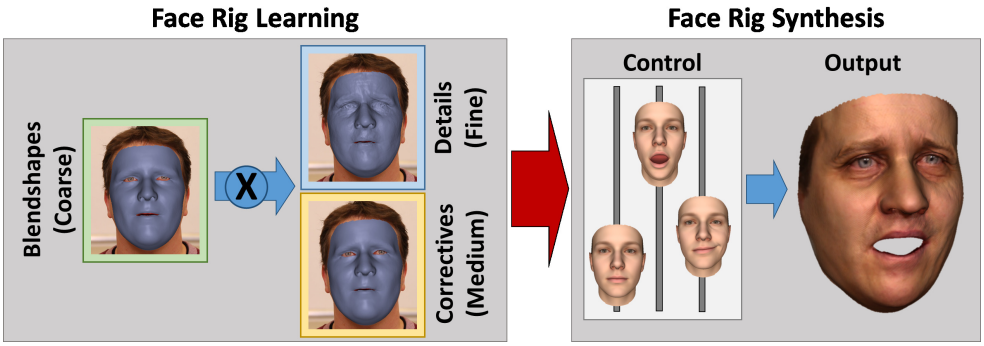


Figure 8.2: Overview. The proposed method learns the coupling between the coarse-scale expression changes (blendshapes) and medium-/fine-scale surface deformations (here for the sequence shown in Figure 7.2) to create a fully controllable 3D face rig that can synthesize new actor-specific detailed expressions by simply modifying blendshape controllers.

with depth cameras [Li et al. 2015b], or also by interpolating dense high-quality scans in a video-driven way [Fyffe et al. 2014]. In contrast, the proposed approach is fully automatic and learns such a model from just arbitrary monocular performance capture data. Note that no manual intervention during face rig creation is required as in [Alexander et al. 2010; Weise et al. 2011]. Moreover, no additional input other than arbitrary performance capture data is needed, i. e., no specific sequence of facial expressions [Ichim et al. 2015; Li et al. 2010] and no face detail regression model learned off-line from a tailored database [Cao et al. 2015].

The first contribution is the automatic extraction of a parametrized rig that models the correlation between coarse-scale blendshape weights and person-specific idiosyncrasies on the medium- and fine-scale detail layer from monocular performance capture data acquired in unconstrained setups. The second contribution is a novel sparse regression approach that exploits the local support of blendshapes to produce more accurate and realistic face rig animations of the medium- and fine-scale layers. This chapter presents captured face rigs for several actors reconstructed from various monocular video feeds, ranging from HD input to vintage video from YouTube. New face animations can be generated with these rigs and they can be used to realistically edit video footage (see Section 8.5).

8.2 Overview

Given an unconstrained video of an actor $\mathcal{F} = \{f_1, \dots, f_T\}$ (e. g., a vintage movie or YouTube video) where T is the total number of frames, the method presented in Chapter 7 provides an estimate of the scene lighting γ and computes a personalized albedo map of the actor's face $\mathbf{C}_p$. In addition, it reconstructs a personalized 3D model $\mathcal{M}$, parametrized on multiple layers: Coarse-scale $\mathcal{M}^{\mathcal{C}}$, medium-scale $\mathcal{M}^{\mathcal{M}}$ and fine-scale $\mathcal{M}^{\mathcal{F}}$. As described in Section 7.3, the shape layers are further parametrized by a sequence of blendshape weights $\Delta_{\mathcal{F}} = \{\boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(T)}\}$, corrective parameters $\mathcal{T}_{\mathcal{F}} = \{\boldsymbol{\tau}^{(1)}, \dots, \boldsymbol{\tau}^{(T)}\}$ and deformation gradients encoding fine-scale surface detail $P_{\mathcal{F}} = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(T)}\}$, respectively. Please refer to Section 7.3 for further details.

To learn and intuitively control a personalized 3D face rig that models not only coarse-scale deformation but also medium- and fine-scale shape detail from the captured data, two main steps are performed, as shown in Figure 8.2:

S1 **Face Rig Learning (Section 8.3):** An optimal affine map $\hat{\mathbf{X}}^{\{M,F\}}$ between the sequence of blendshape weights $\Delta_{\mathcal{F}}$ and each detailed layer $\{\mathcal{T}_{\mathcal{F}}, P_{\mathcal{F}}\}$ is learned separately using a novel sparse regression approach.

S2 **Face Rig Synthesis (Section 8.4):** Given an input set of blendshape weights $\hat{\delta}$ obtained either from blendshape controllers or a tracked performance, the linear maps $\hat{\mathbf{X}}^M$, $\hat{\mathbf{X}}^F$ are then used to predict a medium-scale corrective layer $\hat{\tau}$ and a fine-scale detail layer $\tilde{\mathbf{p}}$, respectively. These layers then enable us to synthesize a new instance of the model $\mathcal{M}$.

The output is a personalized face rig that automatically couples medium- and fine-scale details to intuitive blendshape weights (e. g., represented in the form of sliders), and can generate novel person-specific expressions that preserve the mannerisms and details of a target actor, even when such expressions have not been observed directly in the input data, as shown in Figure 8.2. As the method proposed in Chapter 7 captures a personalized albedo map for the target actor, the face rig can be rendered with photo-realistic face appearance and be used as a 3D avatar to perform facial animation and editing tasks.

8.3 Face Rig Learning

The output of the method presented in Chapter 7 is a personalized parametric 3D model $\mathcal{M}_t$ for each of the T frames f_t that includes a coarse-scale, medium-scale and fine-scale detail layer. While the coarse-scale parametric blendshape model allows for intuitive modification of the rig (e. g., by an artist), there is no equally convenient and semantically meaningful way to create medium- and fine-scale details that match new expressions. This is mainly because the detailed layers do not provide any semantic parametrization nor are they coupled to blendshape expressions. To alleviate this problem, we learn the correlation between blendshapes and the higher detail layers, thus enabling full control of all detail levels by only using the blendshape coefficients.

In the following, a novel sparse and affine regression strategy is presented which learns a mapping between activated blendshape weights and the detail layers, while taking into account the local support of the expression basis.

8.3.1 Affine Parameter Regression of Correctives and Details

Given a sequence of input motion parameters $\Delta_{\mathcal{F}}$ and a corresponding sequence of details $\mathcal{H} \in \{\mathcal{T}_{\mathcal{F}}, P_{\mathcal{F}}\}$, we aim to find an affine mapping to encode their correlation. To this end, the weights of the $K_e = 75$ blendshapes are first stacked in a matrix $\mathbf{W}$:

$$\mathbf{W} = \left[\begin{array}{c|c|c} \delta^{(1)} & \cdots & \delta^{(T)} \\ \hline 1 & \cdots & 1 \end{array}\right] \in \mathbb{R}^{(K_e+1)\times T} \quad . \tag{8.1}$$

Note that the last row of $\mathbf{W}$ implements a constant bias in the estimation that is especially important if certain blendshape weights are not activated in the training set. The detail layer $\mathcal{H}$ is stacked accordingly in a corresponding matrix $\mathbf{H} \in \mathbb{R}^{H\times T}$. It is important to remark that the dimensionality of the parametric fine-scale detail layer is $H = 9J$ (with J the number of mesh triangles), since we regress the per-face deformation gradients. For the medium-scale detail layer, we regress the weights τ, therefore $H = 3K_c = 237$. As a reminder, each per-face deformation $\mathbf{p}_j^{(t)}, \forall j,t$ is encoded

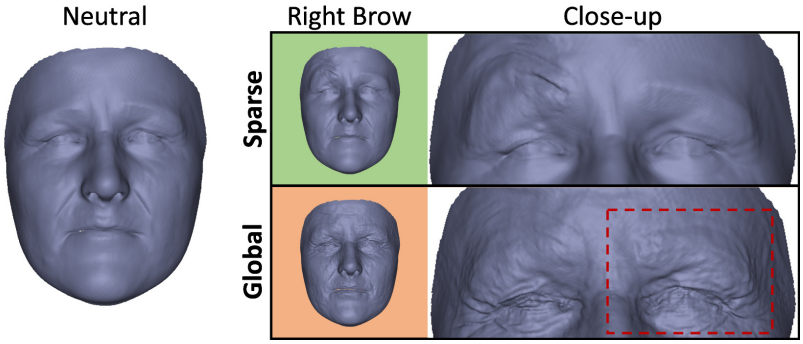


Figure 8.3: Sparse vs. global fine-scale detail prediction. The proposed novel sparse regression formulation (top) obtains more realistic results than global regression (bottom). Note the wrong transient detail around the left eye (red) when the right eyebrow's blendshape is triggered.

by 9 parameters representing the rotation, scaling and shearing of the triangle in the model. Thus, the deformation gradient $\mathbf{p}^{(t)}$ is a $9J$ dimensional vector. Furthermore, the medium-scale corrective layer is represented by a subspace of the $K_c = 80$ lowest-frequency eigenvectors of the Laplace Beltrami operator, which are replicated three times to represent a full 3D deformation basis. However, since the lowest eigenvector has zero eigenvalue, it was discarded from the spectral basis, leading to $K_c = 79$ basis vectors (and therefore $3 \cdot 79 = 237$ corrective parameters).

Our task is to learn an affine mapping $\mathbf{X} \in \mathbb{R}^{H \times (K_e+1)}$ that maps the blendshape weights onto the corresponding details $\mathbf{XW} = \mathbf{H}$. This problem is solved in a least-squares sense by adding a ridge regularizer on $\mathbf{X}$:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \| \mathbf{XW} - \mathbf{H} \|_{\mathcal{F}}^2 + \lambda \| \mathbf{X} \|_{\mathcal{F}}^2 \quad , \tag{8.2}$$

where $|| \, . \, ||_{\mathcal{F}}$ denotes the Frobenius norm and λ is a user-defined ridge parameter that controls the amount of regularization. Such a linear model is known in the literature as ridge regression [Hoerl and Kennard 2000]. Here, the effect of the ridge regularizer is two-fold: 1) It prevents overfitting and 2) it regularizes high-frequency noise due to small tracking inaccuracies which may be introduced by the method proposed in Chapter 7. A closed form least-squares solution for $\hat{\mathbf{X}}$ is given by:

$$\hat{\mathbf{X}} = (\mathbf{W}^\top \mathbf{W} + \lambda \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{H} \ , \tag{8.3}$$

where $\mathbf{I}$ denotes the identity matrix.

8.3.2 Sparse Affine Regression of Fine-scale Details

For the medium-scale layer of correctives ($\mathcal{H} = \mathcal{T}_{\mathcal{F}}$), simple affine regression is sufficient to obtain high-quality results, since the spectral basis has global support. However, the same strategy leads to artifacts when utilized for the prediction of fine-scale surface detail ($\mathcal{H} = P_{\mathcal{F}}$). As shown in Figure 8.3, geometric detail may wrongly appear even if the triggered blendshape does not influence the corresponding surface region. To alleviate this problem, the best affine mapping $\hat{\mathbf{X}}_j^F$ for each triangle $j \in [1 : J]$ is found independently by exploiting the spatial support of the blendshape basis

during training, as follows:

$$\hat{\mathbf{X}}_j^F = \arg\min_{\mathbf{X}_j^F} \| \mathbf{X}_j^F \mathbf{D}_j \mathbf{W} - \mathbf{H}_j \|_{\mathcal{F}}^2 + \lambda \| \mathbf{X}_j^F \|_{\mathcal{F}}^2 \quad , \tag{8.4}$$

where $\mathbf{H}_j = [\mathbf{p}_j^{(1)}, \cdots, \mathbf{p}_j^{(T)}] \in \mathbb{R}^{9\times T}$. The spatial support of the k-th blendshape w. r. t. the j-th triangle is encoded in the diagonal discriminator matrix $\mathbf{D}_j = \text{diag}(d_1^j, \cdots, d_{K_e}^j, 1) \in \mathbb{R}^{(K_e+1)\times(K_e+1)}$. This allows each triangle to switch on or off certain blendshapes based on their influence:

$$d_k^j = \begin{cases} 1 & \text{if } \delta_k \text{ influences the } j\text{-th triangle,} \\ 0 & \text{otherwise.} \end{cases}$$

Due to some outlier support regions in the blendshapes, $K_e = 75$ manually corrected support masks rather than the actual spatial supports were utilized to compute $\mathbf{D}_j$. Note that this correction was performed only once as all the models share the same topology. This novel affine sparse regression strategy for fine-scale details produces superior results, as illustrated in Figure 8.3.

8.4 Face Rig Synthesis

Once learned from data, the linear maps $\hat{\mathbf{X}}^M$, $\hat{\mathbf{X}}^F$ in Section 8.3 are employed to incrementally predict a medium-scale corrective layer $\hat{\tau}$ and a fine-scale detail layer $\tilde{\mathbf{p}}$, which in turn are used to synthesize a medium-scale $\mathcal{M}^M$ and fine-scale $\mathcal{M}^F$ model for new blendshape expressions, accordingly.

8.4.1 Medium-scale Correctives Synthesis

Given new blendshape weights (with 1 appended) $\hat{\delta} \in [0,1]^{K_e+1}$, the medium-scale corrective layer is predicted as $\hat{\tau} = \hat{\mathbf{X}}^M \hat{\delta}$, where $\hat{\mathbf{X}}^M$ is defined as in Equation 8.2 with $\mathcal{H} = \mathcal{T}_{\mathcal{F}}$.

Let us define $\mathcal{P}_n^e(\alpha, \hat{\delta})$ as the coarse deformation field of the coarse-scale model $\mathcal{M}^C$, parametrized by the blendshape weights $\hat{\delta}$ and the identity parameters α. Let us further define $\mathcal{P}^c(\hat{\tau})$ as the corrective deformation field, parametrized by the corrective parameters $\hat{\tau}$ (further details about the parametric model can be found in Section 7.3). Having the predicted medium-scale corrective layer $\hat{\tau}$, we can then reconstruct the corrective deformation field $\mathcal{P}^c(\hat{\tau})$ and apply it on a per-vertex level to the coarse-scale model $\mathcal{M}^C$, yielding $\hat{\mathbf{v}}_n = \mathcal{P}_n^e(\alpha, \hat{\delta}) + \mathcal{P}_n^c(\hat{\tau})$, where $\hat{\mathbf{v}}_n, \forall n \in [1:N]$ denotes the n-th vertex of the medium-scale model $\mathcal{M}^M$ and N is the total number of vertices.

Since the regressed 3D displacements are not rotation invariant, this step is executed in canonical model coordinates.

8.4.2 Fine-scale Detail Variation Synthesis

The high-frequency detail is synthesized on top of the medium-scale result $\hat{\mathbf{v}}_n$, leading to the final embedding $\tilde{\mathbf{v}}_n$.

Given the new blendshape weights $\hat{\delta}$, we predict the detail $\tilde{\mathbf{p}}_j = \hat{\mathbf{X}}_j^F \hat{\delta}$ for the j-th triangle of the parametric model $\mathcal{M}$, where $\hat{\mathbf{X}}_j^F$ is defined as in Equation 8.4. From the 9-dimensional vector $\tilde{\mathbf{p}}_j$,

the per-face affine transformation matrix $\tilde{\mathbf{A}}_j$ can be recovered by inverting the polar decomposition explained in Section 7.3.5. Finally, we use the deformation transfer approach proposed by Sumner and Popović et al. [2004] to augment the medium-scale result with the fine-scale surface detail, yielding an instance of the fine-scale model $\mathcal{M}^F$. For rotation invariance, this transformation is also applied in canonical model coordinates. Note that we neither learned nor regressed fine-scale detail for the surface region inside the eyes, since it is implausible to assume that eye-region deformations can be parametrized via blendshape weights. In view of this, we computed the mean deformation of that region over the entire sequence and kept it fixed in the synthesis.

8.5 Experiments

We demonstrate the applicability of the reconstructed 3D face rigs in various application scenarios that are relevant in facial animation and video editing. We also present qualitative and quantitative evaluations that compare the prediction accuracy of the proposed sparse regression approach with respect to tracked data (Chapter 7) and a state-of-the-art detail synthesis approach.

In total, 9 personalized face rigs were reconstructed from the sequences reported in Section 7.7 and further described in Appendix A (see Section A.1). As a reminder, the test data include: Three indoor sequences (SUBJECT1, SUBJECT2, SUBJECT3), two outdoor sequences (SUBJECT4, SUBJECT5) and four legacy videos (ARNOLD YOUNG, ARNOLD OLD, OBAMA, BRYAN) downloaded from YouTube. The reconstructed face rigs consist of $N = 200$k vertices and $J = 400$k triangle faces, and have an associated personalized albedo map obtained from the method introduced in Chapter 7.

Since most of the results, especially the reconstructed face rigs, can be appreciated best as video, the reader is urged to watch the supplemental videos at the project website[1].

Runtimes The proposed sparse regression approach was implemented on the CPU using simple parallelization routines in OpenMP. Overall, our implementation takes 10 ms for the medium-scale layer and 2 s for the fine-scale detail layer on an Intel Core i7-3770 CPU (3.4 GHz). These runtimes consider both the prediction and synthesis of the different layers. We believe that the computation of the fine-scale detail layer could be massively parallelized on the GPU, thus drastically reducing the computation time.

8.5.1 Application Scenarios

The proposed method automatically creates a fully parametrized 3D face rig of an actor, given as input monocular reconstructions captured from arbitrary footage. As the obtained face rigs are represented on multiple detail layers in a flexible way and can be conveniently controlled by simple blendshape expression parameters, they can be applied in many different application scenarios, e. g., interactive animation, video modification and facial reenactment, which are illustrated in the following.

Interactive Animation To demonstrate the versatility of the new parametric representation, we allow the modification of blendshape parameters via interactive controllers to explore the rig's expression space, as shown Figure 8.4. Note that this application scenario exemplifies, in a simple

[1] http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig/

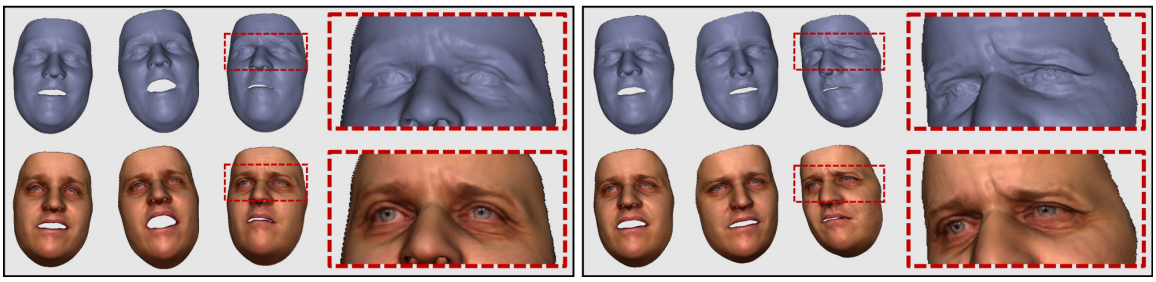


Figure 8.4: Interactive animation - SUBJECT2. Our high-quality parametrized 3D rig allows for the creation of novel and expressive poses of an actor by interactively adapting the corresponding blendshape weights. Here, six different poses with (bottom) and without texture (top) have been interactively generated using an interface with blendshape controllers. Note that the medium- and fine-scale details (top) are automatically predicted using the learned sparse affine regression model.

Figure 8.5: Video modification - ARNOLD YOUNG (left), SUBJECT2 (right). The fine-scale detail layer of both actors is exchanged with the fine-scale layer estimated on ARNOLD OLD for the sequence shown in Figure 7.6. This produces a new synthetic face sequence (bottom) with slight wrinkles that are not part of the original footage (top). Here, the exchanged fine-scale detail layer is dynamically controlled by the expressions (i. e., blendshape weights) tracked in the input sequence of each actor.

way, the task performed by a digital artist in the animation step of the digitization pipeline. The automatically predicted person-specific medium-scale and fine-scale surface detail plausibly matches the new coarse-scale facial expression. Note that these novel expressions are not part of the training set that was used to learn the sparse affine regressor.

More examples of interactive face rig animation can be found in the second supplementary video at the project website[2].

Video Modification As the unconstrained model-based approach presented in Chapter 7 recovers an estimate of the scene lighting as well as the intrinsic and extrinsic camera parameters, we can exploit the potential of our high-quality 3D face rig to photo-realistically modify the face in the original video. To be more precise, we can render a modified face model under the estimated scene lighting and then overlay the correctly rendered and lit face on top of the video. Figure 8.5 shows an example where we exchange the estimated fine-scale detail layer of ARNOLD YOUNG and SUBJECT2 with that of the fine-scale layer learned on ARNOLD OLD (see Figure 7.6) which contains more face wrinkles. By resynthesizing the face rig with Arnold Old's wrinkles and overlaying it on top of the original video, a virtual aging effect can be simulated in video. Figure 8.6 illustrates a more sophisticated video editing example. In this case, we employ the estimated medium-scale and

[2] http://gvv.mpi-inf.mpg.de/projects/PersonalizedFaceRig/

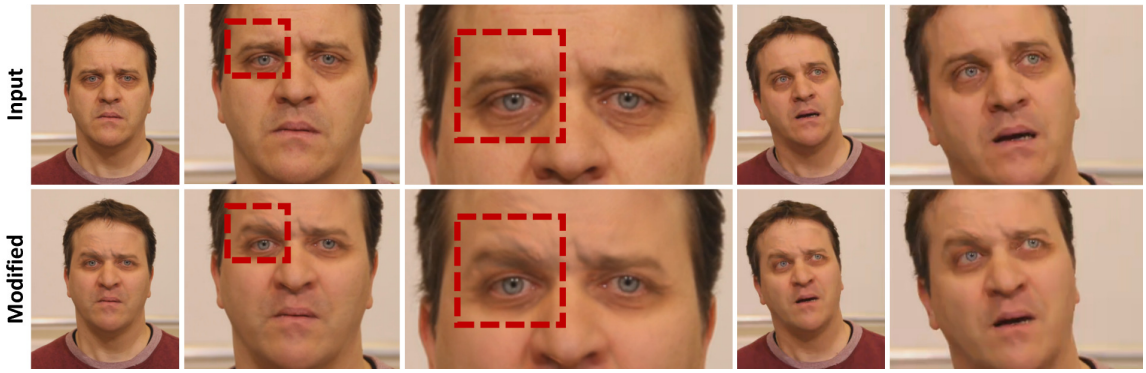


Figure 8.6: Video modification - SUBJECT2. The right eyebrow of the actor is virtually lifted to produce an effect where the actor is playing the same scene in a different/worried mood.

fine-scale detail layer of SUBJECT2 and alter the original expressions of this subject by forcing the right eyebrow to stay up in the whole performance. The modified face rig can then be rendered with the estimated scene illumination over the original video to produce new photo-realistic expressions. Please refer to the supplementary video available at the project website to appreciate both editing effects in motion.

3D Facial Reenactment Since all face rigs are parametrized on the basis of the same blendshape model, we can transfer facial performances between different actors, as shown in Figure 8.7. In this case, the tracked blendshape expressions (i. e., the coarse layer) of SUBJECT2 (source actor) were transferred to the captured face rigs of ARNOLD OLD and OBAMA (target actors). Please note that the person-specific medium-scale and fine-scale detail corresponding to the target actors are inferred for every transferred coarse expression using the sparse regression approach described in Section 8.3, i. e., they are not a mere copy of the personalized layers of the source actor. This leads to more natural and realistic results, since the expression transfer preserves person-specific idiosyncrasies of the target actors. The creation of the rig and the animation is fully automatic and solely based on data captured from a single monocular video sequence, i. e., neither a high-quality face scan [Wu et al. 2016] nor a community photo collection [Suwajanakorn et al. 2014] of the actor has been used in the process.

8.5.2 Validations

Cross-validation and Parameter Tuning To quantify the influence of the ridge regularization term in the estimation of the medium- and fine-scale layer, we compared several regressors learned with different ridge regression parameters λ by measuring the geometric prediction error. To perform the cross-validation, we employed two test sequences (SUBJECT1 and SUBJECT2) and learned a regressor on each sequence for different values of λ. In our experiments, the first half of the tracked sequence was used as training data, while the other half was employed to predict the deformation of the medium-scale layer $\hat{\boldsymbol{\tau}}$ and fine-scale layer $\tilde{\mathbf{p}}$. Note that the prediction error was computed as the Euclidean distance of every predicted 3D vertex position to its corresponding tracked 3D position. The average prediction error of the medium-scale and fine-scale detail layer over the two test sequences can be found in Table 8.1 and Table 8.2, respectively.

As it can be observed in the tables, the lowest prediction error of the medium-scale layer is obtained

Figure 8.7: Facial reenactment. Here, we retarget the rigid head pose and the non-rigid facial motion (coarse expressions) of SUBJECT2 (top row) to the high-quality 3D rigs of two target actors: ARNOLD OLD (middle row) and OBAMA (bottom row). Since the detail layers are regressed for every input expression, the target actor's characteristics are effectively preserved in the reenactment.

Table 8.1: Cross-validation test. Average prediction error (medium-scale).

	Prediction error (in mm)			
Sequence	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 1.5$
SUBJECT1	0.98 ± 0.18	0.96 ± 0.17	0.95 ± 0.17	0.96 ± 0.17
SUBJECT2	0.87 ± 0.17	0.87 ± 0.17	0.87 ± 0.16	0.88 ± 0.16
Overall	0.93 ± 0.18	0.92 ± 0.17	0.91 ± 0.17	0.92 ± 0.17

Table 8.2: Cross-validation test. Average prediction error (fine-scale).

	Prediction error (in mm)			
Sequence	$\lambda = 0.1$	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 1.0$
SUBJECT1	0.30 ± 0.03	0.30 ± 0.03	0.29 ± 0.03	0.29 ± 0.03
SUBJECT2	0.53 ± 0.07	0.53 ± 0.06	0.54 ± 0.06	0.54 ± 0.05
Overall	0.42 ± 0.05	0.42 ± 0.05	0.42 ± 0.05	0.42 ± 0.04

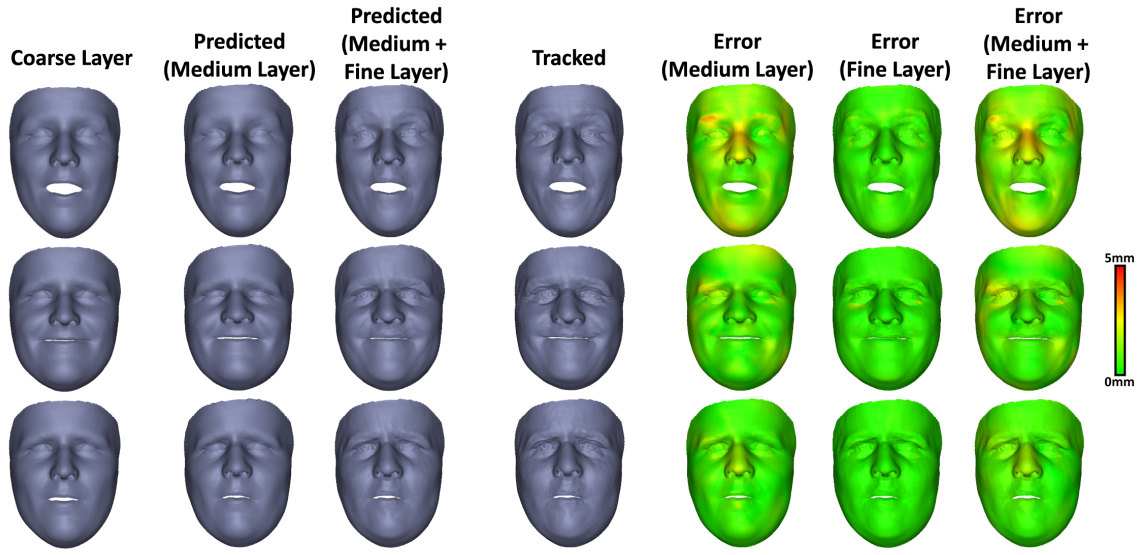


Figure 8.8: Evaluation of the prediction accuracy. Our novel sparse regression strategy infers high-quality medium-scale and fine-scale detail layers given a novel blendshape expression. Note that here we compare quantitatively to the tracked ground truth reconstruction which is accurately reproduced. The geometric prediction error of the medium and fine layer together is always smaller than 3.5 mm (1 mm mean and 0.16 mm standard deviation). The error is mainly explained by residuals in the medium layer, while the error of the fine layer is mostly negligible (< 0.4 mm on average).

when $\lambda = 1.0$ is given. On the other hand, the prediction error of the fine-scale layer stays mostly constant when increasing λ, but a higher regularization tends to over-smooth the results. This means that low values of λ result in more detailed, but slightly noisier predictions due to extrapolation. Empirical experiments showed that the noise is visually negligible for a value of $\lambda = 0.1$. This value achieves good results and is used for all the experiments shown in the thesis.

Prediction Accuracy To evaluate the prediction accuracy, we trained our sparse affine regressor on the first 700 frames of the test sequence (2000 frames in total), and then applied the learned regressor to predict the medium- and fine-scale detail layers on the second half of the sequence. In

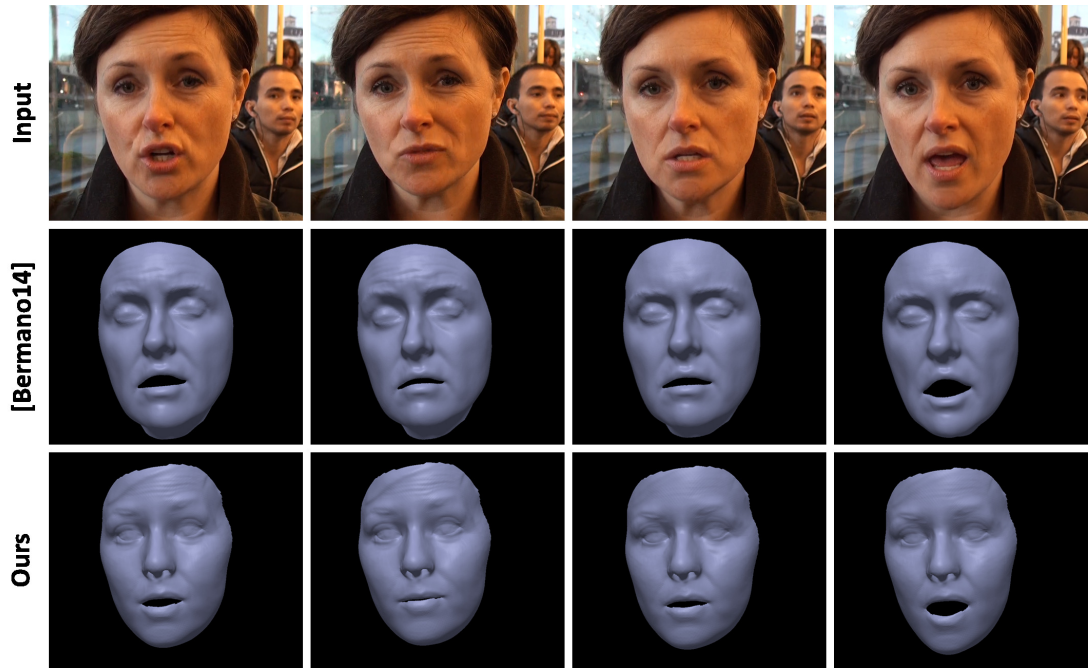


Figure 8.9: Comparison to the state-of-the-art approach by [Bermano et al. 2014] - SUBJECT5. The proposed method (bottom row) obtains predicted correctives and fine-scale detail comparable to Bermano et al.'s method (middle row). However, the latter requires a tailor-made set of training sequences to enhance fine-scale detail and expressiveness.

this experiment, we utilized the values of λ that were found via cross-validation (i. e., $\lambda = 1.0$ for the prediction of the medium layer and $\lambda = 0.1$ for the prediction of the fine-scale detail layer). As ground truth for the comparison, the reconstruction method described in Chapter 7 was run on the complete dataset to get the actually fitted medium-scale and fine-scale layers. Figure 8.8 shows the qualitative and quantitative results. The proposed regressor is able to generalize well beyond the set of expressions used for training.

Comparison to Detail Prediction Methods Our two-layer detail regression approach is compared to the state-of-the-art method by Bermano et al. [2014] for the prediction of actor-specific idiosyncrasies and detail. Figure 8.9 demonstrates that the proposed sparse regression formulation for medium- and fine-scale detail prediction achieves results of comparable quality. Note that Bermano et al.'s method requires a bespoke set of expressive training sequences that are captured with a multiview camera system under controlled lighting from which the fine-scale detail and actor-specific expressiveness are extracted. In contrast, the proposed sparse regression technique was trained using only a subset of frames from the monocular input footage.

8.6 Discussion and Limitations

In this chapter, we have presented the first approach to create a high-quality modifiable 3D face rig of an actor from monocular performance capture data acquired in unconstrained setups. Related to our approach is the recent paper by Ichim et al. [2015] which aims at building a 3D face avatar from video input, but it differs in several ways: First, their approach learns a personalized expression basis

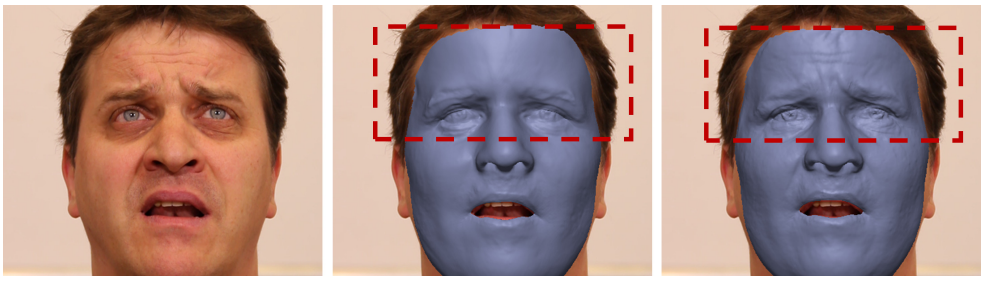


Figure 8.10: Limitations. Even though the monocular model-based approach presented in Chapter 7 tracks actor-specific expressions (right) that accurately match the input data (left), the sparse affine regressor fails to learn and predict the rich space of person-specific nuances and details (middle), mainly when trained only on a short face sequence that does not show much expression variety.

from a specific sequence of captured facial expressions and some steps require manual intervention. In contrast, our approach only needs arbitrary facial expressions of a general unscripted sequence and is fully automatic. Second, Ichim et al. do not learn medium-scale correctives, but optimize the blendshapes themselves. They discuss that learning a full personalized corrective layer, as presented in this chapter, would lead to better personalization.

Despite the high fidelity of the reconstructed face rigs, the proposed data-driven approach relies on the quality of the captured data and therefore shares the limitation of learning-based approaches. Mild occlusions on the face, such as (facial) hair, may be captured as facial features by the method presented in Chapter 7 and wrongly learned as person-specific characteristics, as shown in Figure 8.9. Furthermore, the detail layers are learned based on the correlation to the corresponding expressions observed in the captured data. Thus, we require a sufficient amount of expression variation and detail revelation in the training. If only a short sequence is provided or the actors remain mostly static, their expression space cannot be explored to its full extent. Figure 8.10 illustrates such a limitation. Learning person-specific detailed expressions also requires robust tracking; otherwise, less personalized idiosyncrasies or even geometric tracking drift may be learned, leading to less convincing animations or artifacts in the synthesis. Even though the method introduced in Chapter 7 is quite robust, the reconstruction of highly-deformable surfaces that require accurate depth estimation is still a challenge in unconstrained monocular setups. As a result, our algorithms may not be able to track and learn highly complex mouth deformations, such as kiss shapes and rolling of the lips, which are crucial for photo-realistic 3D avatar animation in movies. Accurate lip tracking from monocular video input is addressed next in Chapter 9. We believe that improvements in this direction will contribute to learning personalized face rigs with very expressive lip shapes.

The reconstructed rigs lack a detailed model for the mouth interior and the (eye) lids. This shares the limitation of related approaches that cannot reconstruct such models from video alone. As such, the rigs were rendered with a static eye albedo map and average 3D eye (lid) shape obtained from the tracked medium-scale and fine-scale layers, as this looks more natural than leaving holes in the eyes. The limitations described above will be further discussed in Chapter 10. Although our detailed rigs have shown themselves to be useful for high-quality animation and editing tasks, they may still fall short of the very high detail and control level required for some professional VFX applications in movies. Even in such cases, our reconstructions could be used by artists as prototypes for customizing rigs or sketching facial animations as well as video editing effects quickly.

8.7 Summary

In this chapter, we have presented an approach for the automatic creation of a fully parametrized, high-quality, and actor-specific 3D face rig from just arbitrary monocular data. The captured rigs are composed of three distinct layers that encode the actor's geometry on all scales: Starting from coarse-scale shape detail up to a layer that accounts for static and transient fine-scale detail. By explicitly learning the correlation between expression variation and the detail layers, a detail prediction model is generated. This enables an intuitive control of the rig based on a small set of control parameters with which artists are familiar. The high fidelity of the reconstructed rigs is shown for several actors from different sources of video, e. g., YouTube footage. As a proof of concept, we have demonstrated the potential of the proposed method for different tasks: Facial animation, expression transfer, and video editing.

The algorithmic improvements proposed in this chapter as well as in Chapter 7 can be considered as a big step towards automatic digitization of fully-controllable, photo-realistic 3D face avatars from unconstrained monocular video input, e. g., legacy footage from feature films. We can anticipate that personalized face avatars will be particularly beneficial for more sophisticated retargeting applications, such as visual dubbing (see Chapter 6) and VR teleconferencing.

To succeed in several applications scenarios, we also require accurate animation of the lip motion to avoid misinterpretation of speech and change of intent. Up to now, our robust tracking algorithms may still fall short of the accuracy we need for tracking complex mouth and lip shapes, mainly due to inherent depth ambiguities and recurrent disocclusions that are hard to resolve from monocular video alone. Improvements in this direction are presented next in Chapter 9.

Chapter 9

Beyond Face Capture: Accurate Lip Tracking

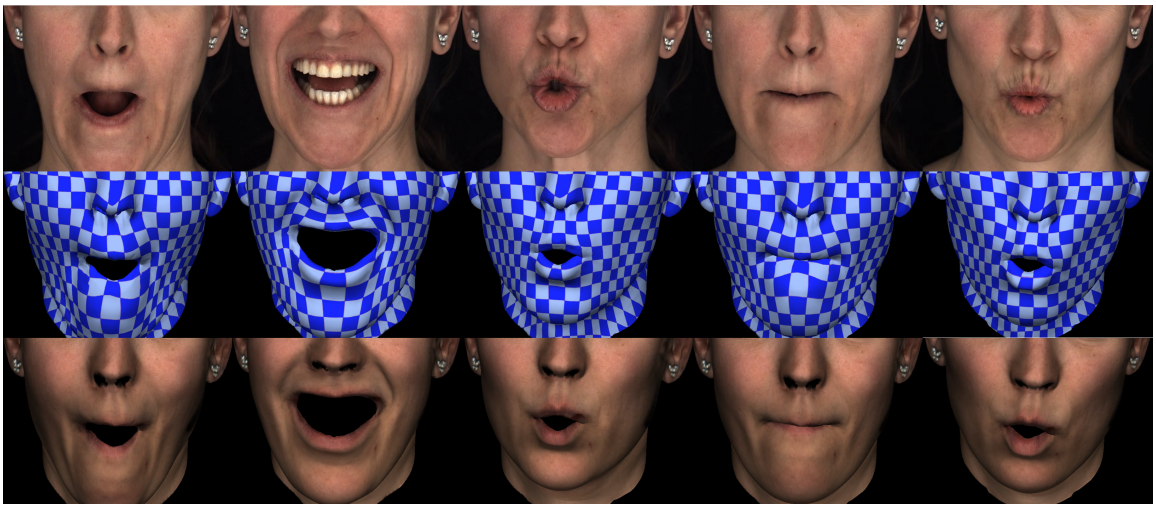


Figure 9.1: Result obtained by the proposed approach - Subject S1. Expressive lip shapes (middle and bottom row), such as kiss and lip rolling, can be reconstructed with high fidelity from just a monocular video (top row).

Accurate capture of shape and motion of the lips is a fundamentally hard problem in facial performance capture for which not many solutions exist (see Chapter 5 and Chapter 7). A solution to it is of paramount relevance in speech recognition and photo-realistic facial animation. This chapter presents a novel robust and versatile regression-based approach for fully automatic reconstruction of detailed and expressive lip shapes, along with the dense geometry of the entire face, from monocular video footage (see Figure 9.1). The method and results presented in this chapter are based on [Garrido et al. 2016b].

9.1 Introduction

When designing virtual avatars, animation artists pay particular attention to the quality and realism of the facial animation. Nowadays, animation artists usually rely on captured facial performances which are used as a baseline to create facial animations, thus drastically simplifying their workflow. Many state-of-the-art methods for high-quality facial performance capture enable dense, static and dynamic reconstruction of the human face from multiview data (see Section 3.1.1). Some methods can now capture the geometry of the entire head, namely the eyelids [Bermano et al. 2015], the eyeball [Bérard et al. 2014], facial hair [Beeler et al. 2012], or scalp hair [Echevarria et al. 2014; Hu et al. 2015; Luo et al. 2012]. More recently, even lightweight methods have been developed that acquire dense face geometry from RGB-D sensors (see Section 3.1.2) or monocular RGB video footage (see Section 3.1.3).

Unfortunately, none of these methods accurately captures the incredible range of shapes and deformations of moving lips. In particular, expressive mouth motions, such as a kiss or expressions with rolling lips, are almost impossible to reconstruct, even with multiview methods in controlled studios. Furthermore, subtle lip shape differences that may disambiguate a friendly smile from a smirk, are very hard to capture. Passive photogrammetric reconstruction of lips is fundamentally hard, since lips are specular and almost featureless in appearance, show subsurface scattering, and exhibit very quick and shape-dependent changes of blood flow. In addition, they are highly deformable (their skin strongly stretches and compresses) and exhibit strong self-occlusions complicating surface tracking. Contour-based tracking is another option to estimate lip shapes. However, while the outer contour of the lips corresponds to a fixed ring on the face, the inner contour is an *occlusion boundary* which is not associated to any fixed location on the lips, making tracking very challenging.

Yet, accurate animation of the lip motion of virtual humans is of paramount importance. Face-to-face communication is multi-modal, i. e., it needs visual and auditory channels. Subtle visible nuances in face and mouth expressions can change interpretation of speech and intent, and exact mouth motion is essential for the hearing impaired relying on lip reading. A video with a purposefully modified lip motion can even make us hear a different consonant – an effect known as the McGurk effect [Nath and Beauchamp 2012]. Thus, animation artists spend a lot of time and effort to adjust incorrectly captured lips.

Only a few passive methods have addressed 3D lip shape reconstruction thus far (see Section 3.2.2). However, most of these approaches require complex professional camera setups and the acquired lip shapes are still limited in deformation range and expressiveness, e. g., lip rolling remains a challenge to capture. Thus, this chapter presents the first automatic method to passively capture detailed expressive lip geometry, along with the dense geometry of the entire face, from just monocular RGB video. The first contribution is the adaptation of a state-of-the-art multiview face performance capture system such that it reconstructs ground truth 3D lip geometry, including rolling and skin stretching. This is accomplished by adding extra lip markers through the application of an artificial color pattern. By using this setup, we can record a training set of high-quality 3D face and mouth motions of several subjects, along with RGB video. The second contribution is a novel model-based face capture method, designed for lip enhancement. At its core is a new regression method based on a radial basis function network trained on the aforementioned database. Our regression method learns the difference between inaccurate lip/mouth shapes found with the multilayer monocular face capture method presented in Chapter 7, and true 3D shapes of lips as well as the surrounding face region reconstructed by the high-quality multiview system. To improve regression accuracy, we additionally use shape features computed from extracted inner and outer lip contours as input to a

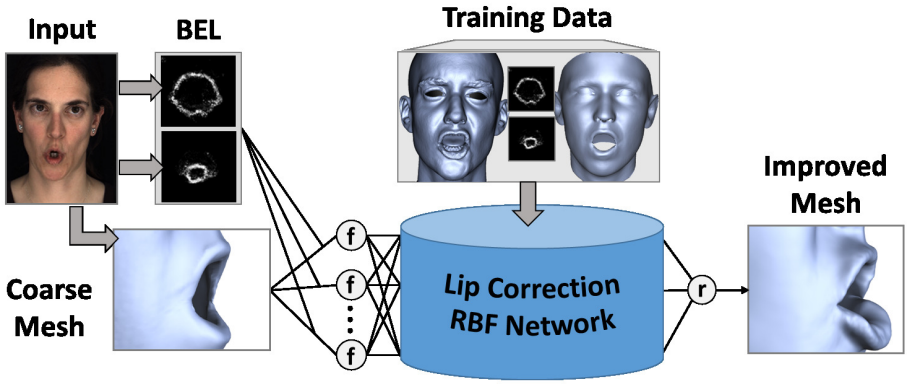


Figure 9.2: Lip shape correction framework. Given a coarse mesh $\mathcal{C}$, and an inner and outer lip contours $\mathcal{B}^I$, $\mathcal{B}^O$ computed by BEL [Dollár et al. 2006], our lip correction RBF network predicts a mesh with improved lips $\mathcal{L}$. The network is trained to learn the difference in geometry between true high-quality mesh deformations $\mathcal{H}$ and inaccurate coarse reconstructions $\mathcal{C}$ by additionally leveraging lip shape features extracted from lip contours $\mathcal{B}$.

robust gradient domain regression strategy.

Quantitative and qualitative results demonstrate that the proposed method can capture complex lip shapes and motions, e. g., protruding lip shapes and lip rolling, at much higher quality than naïve monocular reconstructions. The results also show that our approach generalizes well to unseen individuals and general scenes, enabling high-fidelity reconstruction even from mobile phone videos (see Section 9.6.1).

9.2 Overview

The proposed method takes as input a video of an actor from which a sequence of coarse meshes $\mathcal{C}=\{C_1,\ldots,C_N\}$ with inaccurate lip (and also mouth) shapes is reconstructed, where N is the number of frames in the sequence. At each frame f, inner and outer contours of the lips $\mathcal{B}_f^I$, $\mathcal{B}_f^O$ are also extracted using a Boosted Edge Learning (BEL) algorithm [Dollár et al. 2006]. Given a coarse mesh $\mathcal{C}_f$ and contours $\mathcal{B}_f^I$, $\mathcal{B}_f^O$, our lip shape correction framework estimate a mesh $\mathcal{L}_f$ with improved lip shapes by using a Radial Basis Function (RBF) Network described in Section 9.5.2 (see Figure 9.2). To train the network, three different steps are performed:

S1 **Data collection (Section 9.3):** We create a high-quality lip database and generate the training examples for regression. Our training set consists of high-quality lip shapes $\mathcal{H}$ acquired from a multiview camera system, coarse lip shapes $\mathcal{C}$ obtained using the method described in Chapter 7, and lip contours $\mathcal{B}_I$,$\mathcal{B}_O$ in 2D images detected by BEL.

S2 **Lip correction layer parametrization (Section 9.4):** Next, we establish correspondences between the high-quality reconstructions $\mathcal{H}$ and coarse shapes $\mathcal{C}$, and provide a robust parametrization of the lip shape correction layer based on deformation gradients.

S3 **Lip shape regression (Section 9.5):** Finally, we generate robust features for lip shape correction. These features are used by our RBF network to infer the lip correction layer that allows us to reconstruct high-quality lip shapes $\mathcal{L}$.

(a) (b) (c)

Figure 9.3: High-quality lip database. Using a controlled multiview capture setup (a), and lip tattoos (b), high-quality lip shapes are reconstructed for training (c).

9.3 Data Collection

The main goal is to enhance lightweight face capture methods, in particular by improved reconstruction of the lips (and the adjacent mouth region). Lips tend to be one of the most challenging facial regions, especially for under-constrained capture approaches such as monocular reconstruction. To this end, we construct a training database of high-quality lip shapes and learn a regression function that explicitly maps approximate lip shapes from a lightweight capture method to high-quality and accurate shapes.

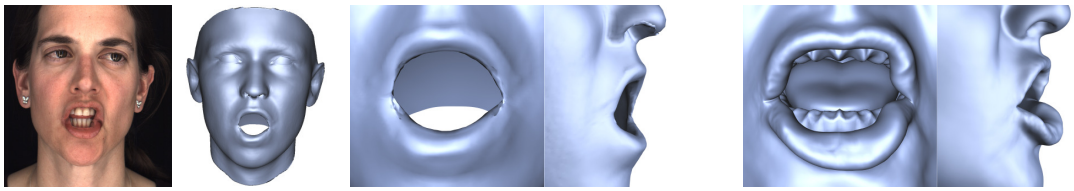
9.3.1 High-quality Lip Database

We build a database of high-resolution 3D lip shapes with the state-of-the-art reconstruction method of Beeler et al. [2011], which uses a multiview camera setup and controlled studio lighting to produce high-resolution face meshes that are in full vertex correspondence over time. For the application at hand, we configure the physical setup such that four cameras are directly focused and zoomed-in onto the lip region (one stereo pair from above and one from below), and six additional cameras (three stereo pairs) frame the entire face, see Figure 9.3 (a). Obtaining highly accurate lip reconstructions even in such a controlled environment can be very challenging, since the lips have very few features and change appearance over time. To overcome this and obtain the best possible 3D data, we apply patterns to the lips via temporary tattoos[1] as shown in Figure 9.3 (b), which provide surface disambiguation and consistency of appearance over time, without drastically altering the natural lip motions of the subject. Figure 9.3 (c) shows a subset of reconstructed lip shapes.

The ground truth training data is cleaned up as a pre-process, e. g., gums are masked out to remove penetrations. We assign correspondences between our base mesh and the ground truth reconstructions once per subject using the method described in Section 9.4.1. No further assignment is needed as both meshes preserve temporal correspondence. Our training set, the first of its kind, contains a very high-resolution and accurate lip shape $\mathcal{H}_f$ for each frame f. The lip shapes span a wide range of lip motions, including smiling, frowning, smirking, kissing, puffing, rolling in/out, sticky lips and side-to-side mouth motions. The dataset consists of both transitions in and out of these complex shapes, as well as general speech animations. The complete database of 3289 total shapes captured from 4 different actors is one of the central contributions in this chapter. To examine the training set, please refer to the second supplementary video at the project website[2]. A comprehensive

[1] www.violentlips.com

[2] http://gvv.mpi-inf.mpg.de/projects/MonLipReconstruction/



(a) (b) (c) (d)

Figure 9.4: Monocular training data. After inpainting the lips (a), we apply the monocular face tracker presented in Chapter 7 (b). The approximate lip shapes are shown in (c), while corresponding high-quality reconstructions are given in (d).

description of the performed lip motions is provided in Appendix B.

9.3.2 Training Data for Regression

Given the acquired high-quality shape $\mathcal{H}_f$ of frame f, we wish to learn the geometric difference between this shape and a coarse monocular approximation $\mathcal{C}_f$. In this chapter, we particularly enhance the lightweight monocular facial tracker described in Chapter 7, but in theory any monocular reconstruction technique could be utilized instead (see, for instance, Chapter 5). Even though the selected monocular method captures dynamic face geometry at state-of-the-art quality (see comparisons in Section 7.7.2), it still struggles to capture expressive lip shapes (see experiments in Section 9.6) like most monocular approaches do.

In order to compute the shape difference for training, we run the lightweight tracker on one of the frontal cameras of the multiview setup. However, the applied lip tattoos would lead to a bias when training the regression function, since during testing the tracker will be applied to monocular data without such artificially added features. To alleviate this problem, we digitally inpaint the sequences to remove the tattoos (details below).

The difference in lip shape between the monocular $\mathcal{C}_f$ and the high-quality reconstructions $\mathcal{H}_f$ can be seen for one pose in Figure 9.4. These differences will be used to train a regression-based lip enhancement algorithm (see Section 9.4). It is important to remark that high-frequency details, e. g., folds and wrinkles, were discarded from both reconstructions and not used for regression, since these features normally represent idiosyncrasies of particular subjects and do not generalize well across different subjects. In our experiments we found that the coarse lip shapes alone are an insufficient feature for robust regression due to the amount of possible ambiguity (see Section 9.6.2). For this reason additional lip contour constraints, detected in the input images using semi-supervised learning, were also incorporated as features (details further down below).

Lip Tattoo Inpainting

Traditional digital inpainting involves replacing corrupt or unwanted image pixels in a semantically meaningful way, typically using surrounding pixels for context. Removing the unwanted lip tattoos is a special case where neither interpolation nor copy operations can generate plausible appearance since the tattoos cover the entire lip region. Fortunately, here we have more information available, namely the reconstructed 3D geometry. We can therefore apply a geometry-guided inpainting process by capturing and reconstructing each actor one additional time without tattoos, and copying the lip region from the un-tattooed image to the tattooed sequences. To this end, we record the

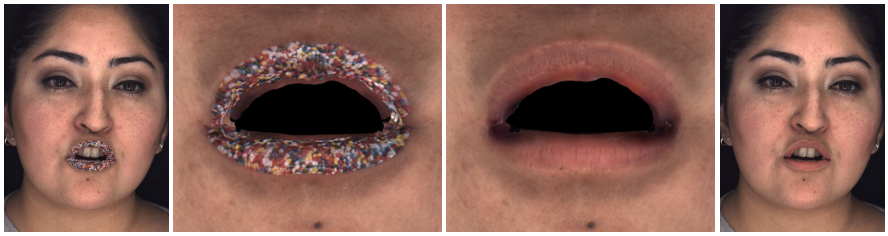


Figure 9.5: Lip inpainting. Geometry-guided inpainting is performed in UV space to remove the lip tattoos. *Left to right:* One image of the training set, the corresponding UV texture, the inpainted UV texture, and the final inpainted image after compositing.

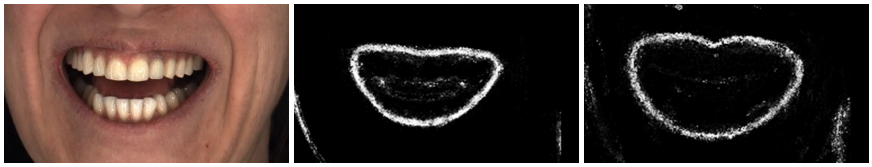


Figure 9.6: BEL contour detection. *Left to right:* Input image, inner lip contour, and outer lip contour.

actor in the high-resolution setup of Beeler et al. [2011] without wearing the lip tattoo and with the mouth slightly open to avoid occlusions. During reconstruction of this pose, we use the same mesh topology as in the lip shape database, putting the un-tattooed shape in a dense vertex correspondence with the training data. We construct a UV texture for the un-tattooed lips by projecting the geometry into the camera images. Then, inpainting each tattooed image can proceed by rendering the lip geometry from the viewpoint of the camera using the un-tattooed texture and compositing the output with the image using a feathering operation at the boundaries.

The drawback of this approach is that the inpainted lips will always exhibit the same appearance and lack dynamic effects such as shape-dependent shading. However, we can compensate for such effects through a shading-equalization scheme. Specifically, we compute the pixel-wise intensity difference of the lip region between each frame and a reference pose, chosen to be similar to the un-tattooed pose, and then add this frame-dependent appearance change to the un-tattooed texture. An example inpainting is shown in Figure 9.5.

Lip Contour Detection

To increase the robustness of the regression function, 2D lip contour features are included in addition to the 3D geometric features. Lips are almost featureless, highly deformable and their appearance changes due to shape-dependent blood flow patterns. The most reliable visual features of the lips are the inner and outer contours, of which the inner is an occluding contour. We employ the BEL algorithm proposed by Dollar et al. [2006] to automatically detect the contours. BEL is a general-purpose supervised learning algorithm that classifies pixels as (non) boundaries over a small image patch based on a large set of generic fast features, including gradients, histograms of filter responses and Haar wavelets at different scales. We train two separate detectors for each different illumination condition (i. e., indoors and outdoors) to regress the inner lip contour $\mathcal{B}_f^I$ and outer lip contour $\mathcal{B}_f^O$ likelihood maps (see Figure 9.6) at each frame f.

In summary, the collected training data includes high quality 3D lip shapes and detected inner and outer 2D lip contours, and corresponding approximate lip shapes from a lightweight face tracker.

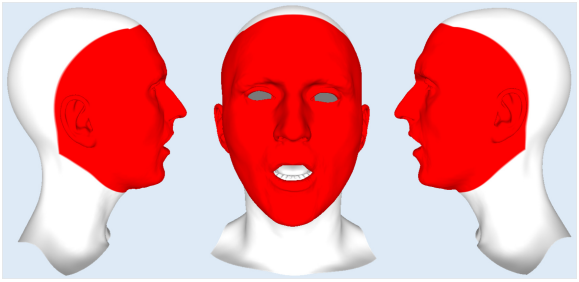


Figure 9.7: Mask used for correspondence association. The red region is employed for computing correspondences between the topology of ground truth and monocular (see Figure 9.8) reconstructions.

9.4 Lip Correction Layer Parametrization

We parametrize the difference between the high-quality reconstructions $\mathcal{H}_f$ and the corresponding coarse monocular reconstructions $\mathcal{C}_f$ in frame f using per-triangle deformation gradients, as proposed in [Sumner and Popović 2004]. Later on, this differential lip correction layer ℓ_f is used to get an improved monocular reconstruction $\mathcal{L}_f$.

9.4.1 Dense Correspondence Association

The 3D reconstructions ($\mathcal{H}_f$ and $\mathcal{C}_f$) obtained by the two different reconstruction approaches are not in vertex correspondence, and in general may not even share the same coordinate system. As a first step, we compute a dense set of triangle-to-triangle correspondences based on a fully automatic Laplacian surface registration technique, described below.

Since the only common element of the two reconstructions is the input image, we use image-based landmarks as constraints for the deformation. To that end, we use a facial landmark tracker [Saragih et al. 2011a] to compute a set of 66 sparse landmarks on the inpainted image of the neutral face. Back-projecting the detected landmarks onto both the coarse mesh and the high-quality mesh provide an initial sparse set of surface correspondences. Based on these constraints, we perform Laplacian surface deformation of the coarse mesh, followed by a dense correspondence search via spatial proximity. Finally, a second Laplacian registration step is performed based on these dense constraints and the resulting alignment is used to establish dense triangle-to-triangle correspondences.

We perform the two-step strategy described above only once per subject. Note that the selected inpainted image is chosen to have the mouth slightly open to avoid potential wrong correspondences in the lip region.

It is important to remark that the ground truth $\mathcal{H}$ and coarse $\mathcal{C}$ reconstructions differ in topology. Each high-quality mesh in $\mathcal{H}$ is a full 3D head surface that also includes the inner mouth geometry (see Figure 9.7), whereas each coarse mesh in $\mathcal{C}$ is mainly a 3D face surface (see Figure 9.8). To further ensure the computation of a valid set of dense correspondences between topologies, we manually define a subregion of the face in the ground truth topology (shown in red in Figure 9.7). This subregion excludes, among others, the tongue and gums.

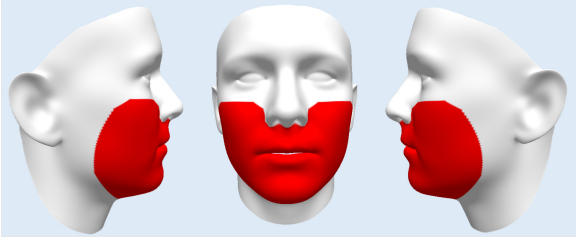


Figure 9.8: Lip correction layer. The lip/mouth region used for regression is shown in red. Note that the whole mesh was utilized for assigning correspondences to high-quality reconstructions (see Section 9.3.2).

9.4.2 Gradient-based Lip Shape Representation

We formulate the shape correction layer in the gradient domain. This is preferable over position-based corrections because the high-quality and coarse meshes may differ by more than just lip shape, e. g., the monocular tracker may also have a slight error in depth, and gradient-based correction is ignorant of such global transformations. The gradient formulation is also advantageous since it allows us to regress improved shapes for only the confined region of the lips and the surrounding mouth area, yet to smoothly blend these improvements with the surrounding face in a subsequent integration step.

The gradient-based lip correction layer captures differences in surface orientation, scale and skew for all the T triangles in the lips and the local mouth region, as defined by the mask shown in Figure 9.8. In a first step, per-triangle deformation gradients $\mathbf{G}_f^{(t)} \in \mathbb{R}^{3\times3}$ between the T faces of the mesh $\mathcal{C}_f$ and the corresponding triangles in $\mathcal{H}_f$ are computed. We map from the monocular tracking results to the high-quality reconstructions using a neutral frame (first frame $f = 0$ of the sequence) as anchor point:

$$\mathbf{G}_f^{(t)} = \underbrace{\mathbf{H}_f^{(t)}}_{\mathcal{H}_0\to\mathcal{H}_f} \cdot \underbrace{\hat{\mathbf{D}}^{(t)}}_{\mathcal{C}_0\to\mathcal{H}_0} \cdot \underbrace{\left[\mathbf{C}_f^{(t)}\right]^{-1}}_{\mathcal{C}_f\to\mathcal{C}_0} . \tag{9.1}$$

The deformation gradients $\mathbf{C}_f^{(t)}$ and $\mathbf{H}_f^{(t)}$ model the expression of the monocular and high-quality reconstruction, respectively. The difference in identity, simply caused by the quality difference in the two trackers, is encoded using $\hat{\mathbf{D}}^{(t)}$. This operator can also account for differences in topology between the reconstructions, for instance, orientation of local mesh triangles. Note that with the proposed deformation gradient operator $\mathbf{G}_f$ only a simple per-triangle mapping needs to be computed. This way, we can avoid any unnecessary deformation transfer step between coarse and high-quality shapes which not only is inefficient, but also deteriorates the quality of the ground truth data.

As already explained in Section 7.3.5, the deformation gradients jointly encode the rotation, scale and shear as a single matrix. This will be problematic for regression, as internally the correction layer will be interpolated linearly.

The deformation gradients jointly encode the rotation, scale and shear as a single matrix. This will be problematic for regression, as internally the correction layer will be interpolated linearly. To overcome this problem, we compute the polar decomposition of the gradient matrix $\mathbf{G}_f^{(t)} = \mathbf{Q}_f^{(t)}\mathbf{S}_f^{(t)}$, factoring into rotation and shear. From these matrices, we extract rotation, skewing and scaling factors (please refer to Section 7.3.5 for further details). In total, this leads to 9 parameters per triangle which allow for linear interpolation. The lip shape correction layer $\ell_f \in \mathbb{R}^{9T}$ stacks

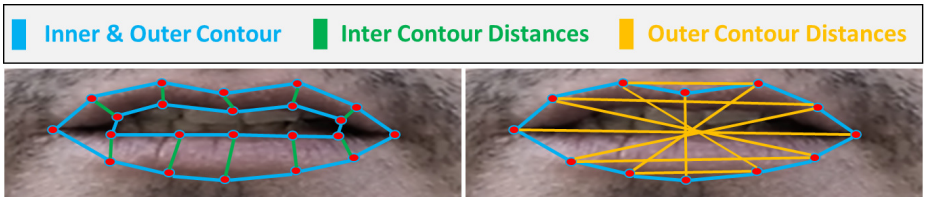


Figure 9.9: Relative distance features. We use inter contour (green) and outer contour (yellow) distances to define the robust features for lip shape regression.

the computed per-face deformation gradients, and will be the target of our regression framework described in the following section.

9.5 Lip Shape Regression

We learn the difference between the inaccurate and true 3D lip shape based on a regression function. The captured training data $\mathcal{T} = \{\mathcal{C}_f, \mathcal{H}_f, \mathcal{B}_f^I, \mathcal{B}_f^O, \ell_f\}_{f=0}^F$ consists of the inaccurate monocular reconstructions $\mathcal{C}_f$, the accurate multiview reconstructions $\mathcal{H}_f$, the computed inner $\mathcal{B}_f^I$ and outer $\mathcal{B}_f^O$ BEL contour maps and the corresponding ground truth output layer ℓ_f.

9.5.1 Robust Features for Lip Shape Regression

We use a set of discriminative features $\mathbf{f}$ that allow us to robustly predict high-quality lip shapes given inaccurate monocular reconstructions. In the feature vector, we jointly encode the inaccurate reconstruction result as well as the target contour constraints. We wish to encode the reconstruction in a compact manner, which is also independent of the particular reconstruction method in order to make the proposed approach as general as possible.

We define a low-dimensional shape subspace ψ by computing Principle Components Analysis (PCA) on the 75 blendshapes used for monocular tracking and keep 99% of the variance. The inaccurate results are then projected onto this subspace to obtain a shape vector of length $|\psi| = 33$.

Target contour constraints are defined by a set of relative features that take the shape of the detected inner and outer lip contour into account. These features are normalized based on the inter-ocular distance, to make the regression results independent of global depth changes. We sample the inner and outer lip contour based on a search that starts from the monocular reconstruction result. To this end, in a pre-process, we specify isolines of the outer lip contour on the template geometry. Starting from sample points on the monocular reconstruction result of the outer contour, we search for the closest maxima in the BEL likelihood maps along the gradient of the isolines. The found maxima in the maps $\mathcal{B}_f^I$ and $\mathcal{B}_f^O$ are the corresponding points of the inner and outer contour, respectively. We use the obtained outer and inner contour points to define a set of relative features that encode 10 distances on the outer contour and 10 distances between the two contours, see Figure 9.9. Note that this exploits the correlation between the 2D contours and the actual 3D lip shape. Overall, together with the PCA coefficients, the lip feature vector $\mathbf{f}$ has $M = |\psi| + 20 = 53$ components.

9.5.2 Local Radial Basis Function Networks

Given the per-frame lip features and corresponding lip correction layers $\{\mathbf{f}_f, \ell_f\}_{f=1}^{F}$, we learn for each of the T triangles of the lip correction layer a regression function $r^{(t)} : \mathbb{R}^M \rightarrow \mathbb{R}^9$ using a vector-valued radial basis function network (RBFN). We use a network architecture with a single hidden layer (see [Bishop 2006]), and the associated $N \ll F$ neurons $\Phi_n : \mathbb{R}^M \rightarrow \mathbb{R}$ have fixed prototypes $\mathbf{p}_n \in \mathbb{R}^M$ in feature space and share the same scale $\beta \in \mathbb{R}$:

$$\Phi_n(\mathbf{f}) = \exp\left(-\beta \cdot ||\mathbf{p}_n - \mathbf{f}||_2^2\right) \quad . \tag{9.2}$$

Prototypes $\mathbf{p}_n$ are obtained by a temporally uniform sampling of the training sequences. The output node implements a linear weighted summation of the per-neuron activation levels and adds a constant bias parameter $\mathbf{b} \in \mathbb{R}^9$:

$$r^{(t)}(\mathbf{f}) = \left[\sum_{n=1}^{N} \mathbf{w}_n^{(t)} \Phi_n(\mathbf{f})\right] + \mathbf{b} \quad . \tag{9.3}$$

We tackle the problem of finding the N weights $\mathbf{w}_n^{(t)} \in \mathbb{R}^9$ for each triangle t using ridge regression [Hoerl and Kennard 2000]:

$$\min_{\{\mathbf{w}_n^{(t)}\}_{n=1}^{N}} \left[\underbrace{\sum_{f=1}^{F} \left\| r^{(t)}(\mathbf{f}_f) - \ell_f^{(t)} \right\|_2^2}_{E_{data}} + \alpha \cdot \underbrace{\sum_{n=1}^{N} \left\| \mathbf{w}_n^{(t)} \right\|_2^2}_{E_{reg}} \right] \quad . \tag{9.4}$$

Here, the data term E_{data} encodes how well the training data is reproduced and the ridge regularizer E_{reg} prevents overfitting. The importance of the regularizer is controlled by the ridge parameter α. Optimal values for α and the scale parameter β are found via cross-validation. Since the optimization problem is quadratic, the minimizer can be found by solving a linear system. Note, the linear system decomposes into 9 independent linear subproblems of size $N \times N$. Since all subproblems share the same system matrix (only the right-hand sides differ), the regression function can be efficiently computed.

Given a new input feature vector $\hat{\mathbf{f}}$, the corresponding per-triangle correction can be obtained by $\hat{\ell}^{(t)} = r^{(t)}(\hat{\mathbf{f}})$. Afterwards, the high-quality lip shape $\hat{\mathcal{L}}$ can be reconstructed by integrating the per-triangle deformation fields using deformation transfer (further details can be found in Section 8.4.2). Note that all steps are performed in a canonical frame for rotation and translation invariance.

9.6 Experiments

We demonstrate the applicability of the proposed method on a variety of different datasets. In total, we captured 3 female and 3 male subjects, henceforth referred to as S1-S6. S1-S5 were recorded indoors in the multiview setup described in Section 9.3.1 using 4MP machine vision cameras, but only S1-S4 were used for training. We additionally captured S3, S4 and S6 outdoors with an iPhone camera (resolution 1920×1080, 30 fps). Finally, we also tried our approach on a legacy video downloaded from YouTube, where the US president (Barack Obama) commemorates Independence

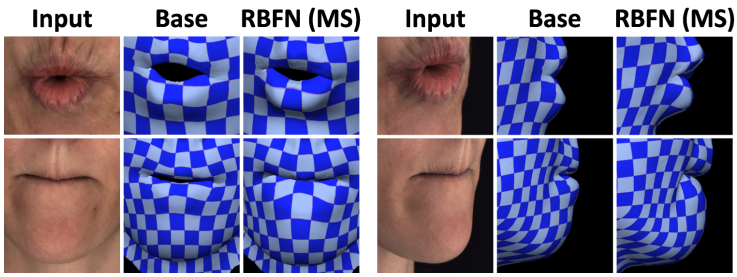


Figure 9.10: Reconstruction quality - Subject S1. The proposed RBFN regression successfully improves the coarse base tracker (Chapter 7) to better handle stretching, bending and rolling of lips.

Day on July 4 (this sequence was also employed to generate the results shown in Section 7.7.1 and Section 8.5.1). Overall, the proposed approach was evaluated on 11 sequences (4 captured using a controlled studio setup and 7 in a general uncontrolled environment).

For the experiments shown below, we evaluate different design choices and compare with a model-based tracking approach using enhanced lip blendshapes and explicit lip contour alignment constraints. In the different quantitative and qualitative evaluations, we use three different types of regressors:

- **PS:** A person-specific regressor trained for a specific subject. This regressor is only applied to sequences of the same subject. Note though that training and testing datasets are disjunct.
- **MS:** A multi-person regressor trained on four different subjects (S1-S4). The test subject can be any of the four. Again, training and testing datasets are disjunct.
- **GR:** A generalization regressor trained on three or four subjects (out of S1-S4). The identity of the test subject is not included in the training set.

The results reported here, especially the motion of the lips, are appreciated best as video. Thus, the reader is strongly advised to watch the supplemental videos at the project website[3].

Runtimes and Parameters On average, the runtime of our method (after training) is approximately 25 sec/frame on an Intel Xeon E5-2637 CPU (3.5 Ghz), where 20 seconds are spent on monocular tracking (Chapter 7) and 5 seconds are added for our new lip correction approach. In all performed experiments, we use every tenth frame of the training set to define the prototype vectors of our RBF lip correction network. All parameters of our regressor remain constant during the experiments (see Section 9.6.2). Note that the lip correction layer modifies $T = 32$k triangles faces, which mainly correspond to the lip and mouth region of the coarse mesh.

9.6.1 Results

We use our novel lip correction network to improve the reconstruction quality of the monocular face tracker presented in Chapter 7, to which we will refer to as coarse base tracker in the following. To this end, we use data captured by one of the frontal cameras of the multiview setup, as well as outdoor video footage captured under general uncontrolled illumination with an iPhone camera. A

[3] http://gvv.mpi-inf.mpg.de/projects/MonLipReconstruction/

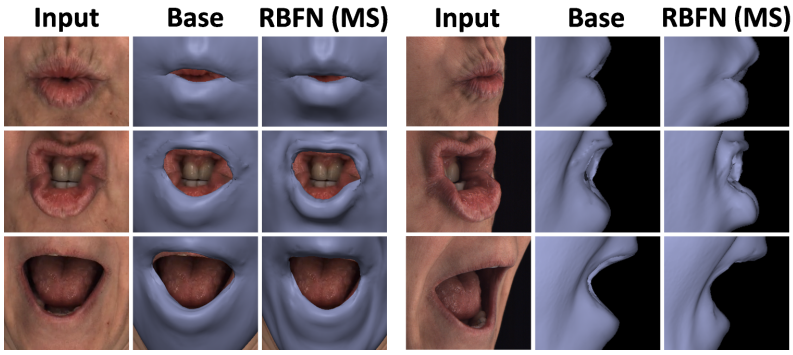


Figure 9.11: Lip protrusion - Subject S1. The proposed regressor is more resilient to the depth ambiguity inherently present in monocular tracking and can plausibly reconstruct protruding and rolling lips.

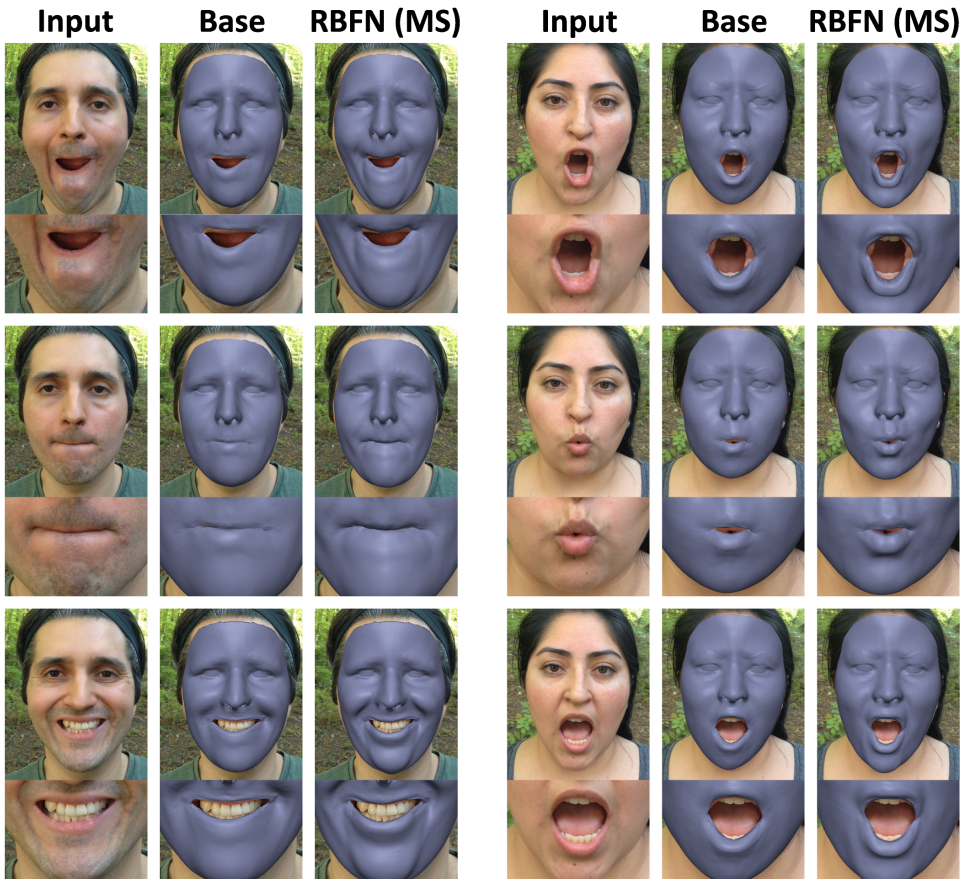


Figure 9.12: Results on outdoor scenes - Subjects: S3 (left), S4 (right). Our proposed regression framework substantially improves on the lip shapes reconstructed by the coarse base tracker (Chapter 7). Note how especially challenging lip motions, such as rolling or stretching, are better captured in the refined results. The regressor is even able to improve the reconstruction quality of the surrounding area, such as nasolabial folds or the chin.

coarse base reconstruction is obtained and the regressed lip correction layer is applied. Figure 9.1 and Figure 9.12 show the coarse base and refined reconstructions for indoor and outdoor setups, respectively. In this experiment, we used the **MS** regressor specified above. As it can be seen, the

Figure 9.13: Generalization results to novel subjects - Subjects: S5 (left), S6 (right). Our generalized regressor **GR** generalizes well to novel subjects and general outdoor scenarios.

regressor successfully improves the lip shapes of the monocular base reconstruction. Especially, inward and outward rolling of the lips, lip protrusions, and the kiss shape are nicely captured. This is further emphasized in Figure 9.11 and Figure 9.10, which visualize surface stretching and shape change from side views.

Please further note that shape improvements are also visible in the face region surrounding the lips,

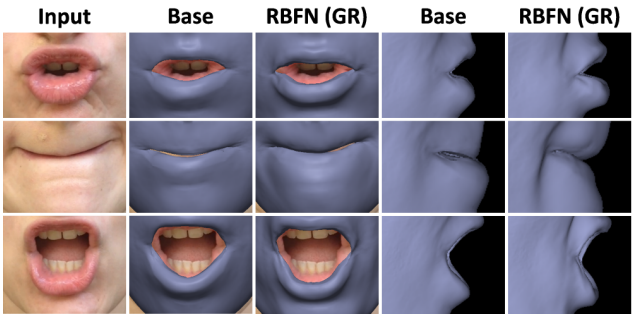


Figure 9.14: 3D lip shape - Subject S6. Our generalized regressor **GR** nicely captures inward and outward rolling of lips (even for novel subjects) given just a monocular video.

Table 9.1: Quantitative evaluation of different reconstruction strategies. Reconstruction error in cm, computed as the average Euclidean distance to the ground truth lip reconstruction $\mathcal{H}$. The reconstruction strategies are (from left to right): Base tracker, base tracker with contour alignment constraints, personalized regressor, multi-person regressor, and generalized regressor.

	Base	**Contour**	**RBFN (PS)**	**RBFN (MS)**	**RBFN (GR)**
mean	0.40	0.39	0.33	0.30	0.32
std. dev.	0.14	0.12	0.12	0.10	0.11

where the regression adds plausible bulging and folding of the skin, which supports the lip shapes (Figure 9.1, left; Figure 9.12, top left). In addition, we applied our generalized regressor **GR** to novel subjects captured both indoors and outdoors; the latter recorded under conditions that substantially differ from the training environment, as shown in Figure 9.13. As it can be observed, the lip correction network generalizes nicely to uncontrolled scenarios and different illumination conditions, since the shape-based features used for regression are less sensitive to changing environment conditions than photometric cues. Again, inspecting the lips closely and from the side (Figure 9.14) clearly illustrates how the overall shape is improved by our regression strategy.

We also applied our approach to an uncontrolled sequence downloaded from YouTube, where the US president speaks naturally in front of the camera[4]. Figure 9.15 shows that our approach also generalizes well to unconstrained capture setups exhibiting some mild head motion and rotation. Note that the regression-based lip correction approach improves the shape of the reconstructed lips (even for blurry images) and captures details around the lips, e. g., the nasolabial folds and dimples.

9.6.2 Validations

Generalization Properties of the Regressor We evaluate the generalization properties of the proposed lip correction RBF network. To this end, we trained a person-specific regressor **PS**, a multi-person regressor **MS** and a generalization regressor **GR**. We qualitatively and quantitatively evaluate the accuracy of these three architectures on a test sequence of S2. Figure 9.16 shows color coded error maps with respect to ground truth reconstructions. For the corresponding numbers see Table 9.1. The obtained reconstruction quality is largely independent of the regressor type. This shows that our approach generalizes well to novel subjects and does not require person-specific training data.

[4] https://youtu.be/d-VaUaTF3_k

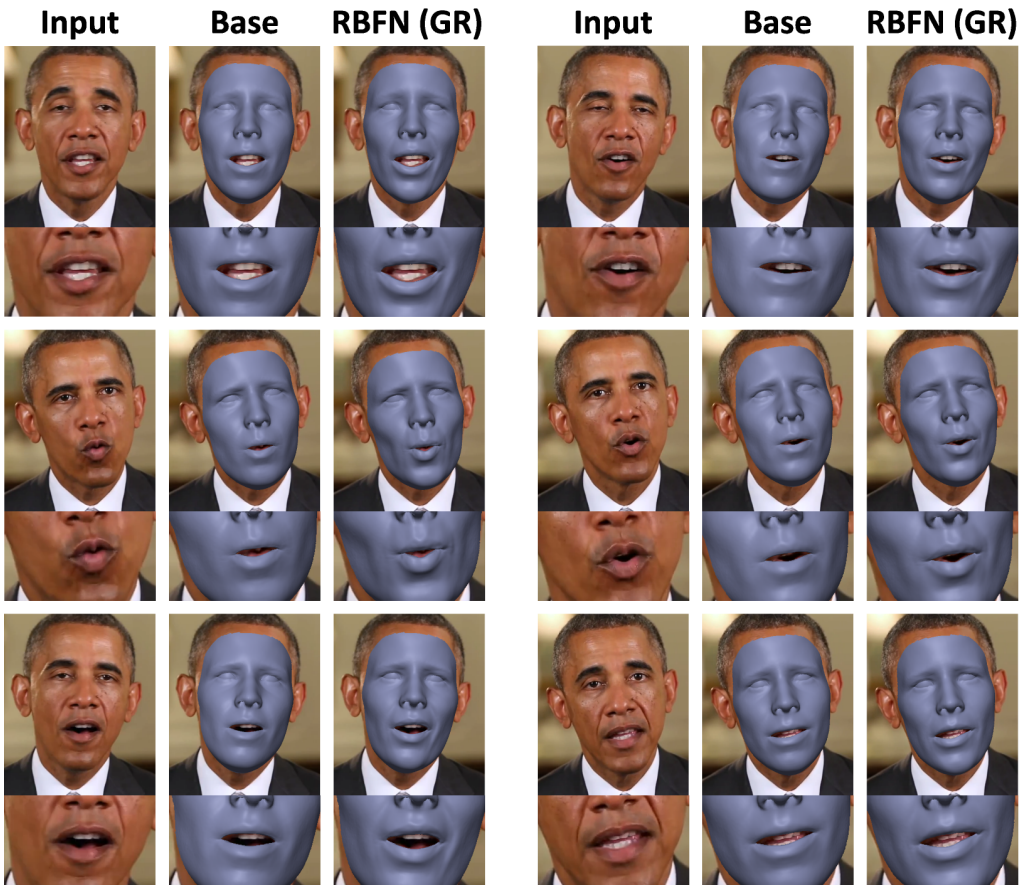


Figure 9.15: Results on internet video footage - Barack Obama. The proposed regression framework generalizes well, even to internet video with general unknown lighting. Compared to the coarse base tracker, our approach reconstructs higher quality lip shapes (even fast lip motions are significantly improved). Note that nasolabial folds and dimples are also corrected.

Evaluation of the Regression Strategy We compare our RBFN regressor with a simple linear affine regressor just as it was used in Chapter 8 for fine-scale detail deformation, see Figure 9.17. Especially surface dynamics are better handled by the proposed non-linear approach. We also quantitatively show this improvement in a cross-validation experiment. To this end, we train both regressors on the same training data, while leaving out a set of validation clips (732 frames). In a first step, we select the best parameters for both regressors using cross-validation. The RBF network performs best for $\alpha = 0.1$ and $\beta = 0.1$. For the linear affine regressor, the Tikhonov regularization parameter $\alpha = 2.0$ leads to the best results. With these parameters, our RBFN regressor obtains an average feature space error of 0.13 (0.04 standard deviation). In contrast, the linear affine regressor has a higher average feature error of 0.14 (0.05 standard deviation).

Influence of Input Features on Regression We also quantitatively evaluate the influence of different input features. To this end, we compare the cross-validation error as well as the tracking error on our ground truth sequence of S2 for different feature descriptors. Table 9.2 and Table 9.3 show that the use of both PCA coefficients and relative distance features improves upon descriptors that are only based on one of these two features. This can mainly be ascribed to certain ambiguities that cannot always be resolved by relative distances or lip shape geometry alone, e. g., symmetrically-

Figure 9.16: Comparison of generalization properties of different regressors - Subject S2. *From bottom to top:* **GR** regressor performs comparable to **MS** and **PS**, and all the regressors improve upon the reconstructions of the coarse base method. This is demonstrated by the heatmap overlays, which show the Euclidean distance error to the ground truth reconstructions $\mathcal{H}$.

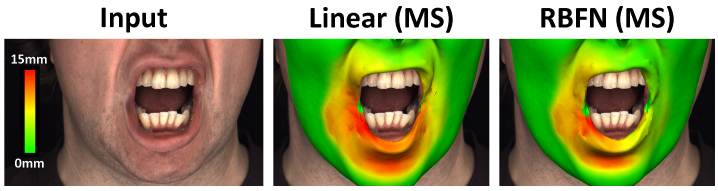


Figure 9.17: Comparison between linear affine regression (middle) and the proposed RBFN regressor (right) - Subject S2. The Euclidean distance between the regressed reconstructions and the ground truth (shown as a heatmap overlay) confirms that non-linear regression leads to smaller errors.

Table 9.2: Influence of features. Cross-validation error in feature space (i. e., deformation gradients space), computed as the average L_2 norm of the difference between the true lip shape correction ℓ and predicted correction $\hat{\ell}$.

	PCA coefficients	Relative distance features	Combined
mean	0.14	0.17	0.13
std. dev.	0.05	0.06	0.04

Table 9.3: Influence of features. Reconstruction error in cm, computed as the average Euclidean distance between the ground truth lip reconstruction $\mathcal{H}$ and the improved lip shapes $\mathcal{L}$.

	PCA coefficients	Relative distance features	Combined
mean	0.32	0.33	0.30
std. dev.	0.12	0.13	0.10

consistent lip deformations or depth-involving lip shapes, respectively.

Comparison to Model-based Lip Tracking In order to perform a baseline comparison, we extended the monocular face tracker presented in Chapter 7, which also serves as our base tracker, by incorporating explicit lip blendshapes and lip contour alignment constraints. Person-specific lip blendshapes have been computed based on the high-quality multiview reconstructions. In particular, we transferred 30 user-selected expressive lip shapes to the 3D identity shape estimated by the monocular tracker using deformation transfer [Sumner and Popović 2004]. Lip contours are detected using BEL and the optimization process tries to align the inner and outer contour of the model with the ridges in the likelihood maps. Since the inner contour is an occluding one, we perform this optimization in an iterative flip-flop fashion. This is similar to the approach used in [Anderson et al. 2013a].

As can be seen in Figure 9.18, the regression-based approach obtains higher quality results, which align to the ground truth better. Table 9.1 shows the corresponding statistics for the reconstruction errors. This test shows that, in particular for capturing the true rolling and stretching of the lips, even enhancing previous model-based methods with additional image constraints is not sufficient. We can obtain a better spatial alignment, more expressive lip shapes and better recover stretching and bending of the lips, without having to tediously augment a parametric 3D expression model per person.

Figure 9.18: Comparison between the proposed RBFN regressor (bottom row) and an augmented model-based tracking (middle row) - Subject S2. The heatmap overlays show that the RBFN regressor outperforms a model-based tracker which utilizes explicit contour alignment constraints and additional person-specific lip blendshapes.

9.7 Discussion and Limitations

In this chapter, we have demonstrated high-quality lip shape reconstructions even for challenging and expressive mouth motions, such as a kiss or rolling lips. While these compelling results are obtained from just monocular video footage, the proposed approach still has some limitations: First, it is based on a set of collected training data and shares the limitation of learning-based approaches, i. e., it does not generalize well to situations that are drastically outside the span of training examples, such as faster or more expressive motions than those used for training. This could be solved by capturing more training data. Such an extension requires the availability of a high-quality multiview setup; however, it is a one time investment.

Furthermore, while the employed feature descriptor is translation invariant, it is not invariant to rotations, especially out-of-plane head orientations. Handling different head rotations would require an extensive amount of additional training data. Alternatively, this problem could be alleviated by compensating for rigid head motion before feature computation, i. e., projecting the sample points onto a tracked plane in front of the mesh and computing the contour distances in 3D. It is also important to remark that BEL is sensitive to lighting and strong color changes. Hence, the detector must be re-trained for the individual illumination conditions (this could in theory be overcome with a sufficiently large dataset containing these variations). By choice, one could also use a different contour detection strategy that is more robust to these variations, which is totally feasible since our algorithm does not directly depend on the chosen structure detector. Drastic appearance changes (e. g., dark vs. pale skin color or beards) could be handled in a similar manner. On the contrary, our shape features are only based on geometric properties and are therefore invariant to these situations.

Mild facial hair is normally captured as high-frequency detail by both the multiview reconstruction and the baseline algorithm. As such, it can be decoupled from the coarse lip motion estimation and does not pose a problem. Thick beards and occlusions, on the other hand, can make the approach fail, since a robust detection of the lip contours would not always be possible. Overall, it is expected that the reconstruction quality can be further improved by increasing the amount of training examples.

As demonstrated by the results in Section 9.6, the proposed lip correction method is capable of regressing the lip shape very well, but since all the features are translation invariant, an accurate alignment to the input data cannot be guaranteed. In many applications this is not of paramount

importance, i. e., lip reading or movie dubbing. Future work will address this by incorporating the detected contours as reprojection constraints into the gradient based reconstruction strategy.

9.8 Summary

In this chapter, we have presented an approach to fully automatically reconstruct expressive lip shapes along with dense geometry of the entire face, from just monocular RGB data. At the core of this approach is a novel robust regression function that learns the difference between inaccurate lip shapes and true 3D lip shapes based on a captured database of high and low quality reconstructions. Rather than resorting to unreliable photometric features, the proposed method utilizes shape features computed from extracted inner and outer lip contours. Qualitative and quantitative results have demonstrated that the proposed monocular approach reconstructs higher quality lip shapes, even for lip rolling or kiss shapes, than previous monocular approaches.

Since subtle visible nuances in face and mouth expression strongly influence the interpretation of speech and intent, we can anticipate that the approach presented in this chapter will be particularly helpful for applications that deal with audiovisual content, i. e., movie dubbing (see Chapter 6) and lip reading.

The algorithmic contributions proposed in this chapter greatly advance the state of the art in photo-realistic facial animation. We believe that when combined with the contributions presented in previous chapters it will be possible to create and animate photo-realistic 3D face avatars with very expressive personalized faces and mouth motions using arbitrary monocular video footage, either captured on set, self-recorded, or even downloaded from the Internet.

There are still some challenges concerning the capture of faces that need to be addressed first to enable full digitization of 3D avatars *in the wild*. Chapter 10 will give an outlook to future directions in this regard.

Chapter 10

Conclusion

Cutting-edge advances in technology in the digitization pipeline now allow the movie industry to create and animate virtual 3D face avatars with personalized expressions that look indistinguishable from real actors. Digitizing photo-realistic faces, however, comes at the price of extensive manual work and sophisticated multi-sensor capture systems that are expensive to build and that only work with in-studio controlled illumination. They also cannot be used for actors that are physically not present, e. g., vintage movie stars. Lightweight approaches have simplified and democratized the capture process by using commodity sensors, but the reconstructed 3D models lack the amount of detail and realism which is necessary to produce video-realistic animations.

We envisage an automatic, lightweight framework that takes the best of both worlds: It creates a photo-realistic, fully-controllable 3D face avatar that can be used for performing complex facial animation and video-realistic editing tasks while being obtained from a 2D video recorded under uncontrolled setups. This way, we could revive vintage actors or even animate ourselves.

This thesis has presented a state-of-the-art toolbox of algorithms towards that goal: Photo-realistic capture, animation, and editing of high-quality synthetic 3D face models from unconstrained 2D video and that are affordable for anyone. Novel technical advances have been developed in three different areas, as briefly sketched in Figure 10.1. On the capture side, we have developed accurate and robust face tracking algorithms, from keyframe-based landmark location refinement and semi-constrained model-based 3D capture to multilayer parametric 3D tracking and regression-based 3D lip reconstruction in unconstrained video footage. On the animation side, we have started with 2D video-based retargeting based on robust image metrics, followed by performance-driven transfer of detailed 3D models. Finally, we have automatically created personalized, controllable 3D rigs to improve retargeting. On the editing side, we have moved from simple image-based compositing to capture-based editing using photo-realistic face albedo and plausible mouth interior synthesis. As a proof of concept, we have tested our methods on different real-life application scenarios: simple texture editing, reenactment, visual dubbing, and video rewriting.

In the following, we conclude the efforts achieved so far. First, we restate and discuss the presented contributions. Then, we review some extensions not explored in this thesis, but recently developed as joint work. Finally, we examine some open challenges towards full head digitization *in the wild*.

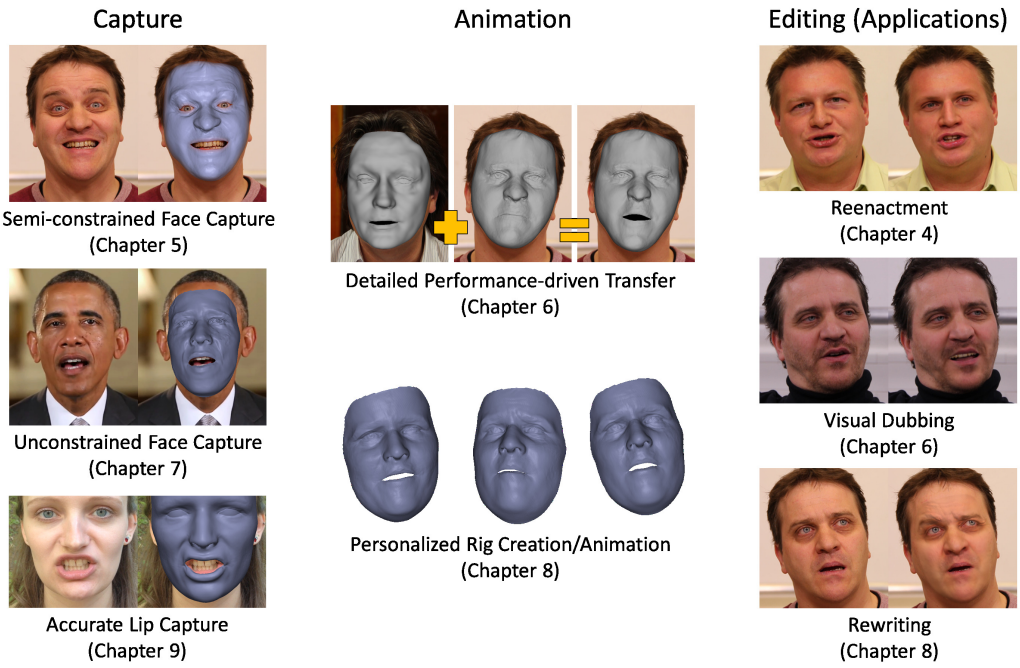


Figure 10.1: Overview of the main contributions presented in this thesis. *From left to right:* Capture: Semi-constrained model-based 3D face capture, personalized multilayer 3D face reconstruction in unconstrained setups, and accurate 3D lip shape regression. Animation: Performance-driven transfer of detailed 3D models, and animation of personalized, fully-controllable 3D face rigs. Editing (applications): Video-based retargeting and simple compositing (reenactment), plausible mouth synthesis aligned to audio (visual dubbing), and photo-realistic face synthesis (visual dubbing, rewriting). Note that the edited results are shown on the right side.

10.1 Summary and Discussion

This section recapitulates the core contributions presented in Chapters 4–9 and provides a brief discussion of some remaining challenges that were not addressed in this thesis.

Chapter 4 presents a fully-automatic, video-based reenactment method that replaces the face of a target actor with that of a user, while preserving the facial expressions of the target actor. At the heart of the approach is an improved 2D tracking algorithm that exploits optical cues between keyframes to track accurate landmarks, which in turn are used to detect and replace faces. Expression transfer is formulated as a retrieval problem that selects source frames based on robust appearance and motion descriptors as well as temporal clustering. Face replacement is performed using a simple, yet effective, warping strategy that preserves facial shape while matching head pose.

Image-based tracking and transfer methods usually exhibit problems in the presence of challenging facial motion and head rotations. To overcome these limitations, Chapter 5 introduces a state-of-the-art model-based approach that captures detailed, spatio-temporally coherent 3D face geometry as well as the incident illumination from 2D videos with known camera intrinsics and coarse 3D geometry of the actor's face. This approach leverages robust landmark tracking (Chapter 4) and temporally-coherent dense optical cues to track the actor's facial motion accurately on long sequences. An adapted shape-from-shading framework [Valgaerts et al. 2012b] allows us to recover the scene lighting and fine-scale skin geometry by exploiting shading cues in the temporal domain.

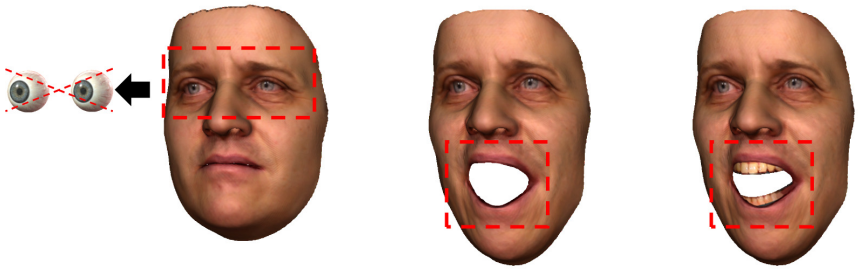
Chapter 6 exploits the potential of the previous model-based approach and introduces a performance-driven system for video-realistic retargeting of detailed facial models whose mouth motion aligns to a new audio signal to perform visual dubbing. Unlike Chapter 4, the mouth motion transfer is conveniently carried out in the blendshape space. A new spatio-temporal rearrangement strategy that employs both the actor's and the dubber's performance then allows us to retrieve a temporally-consistent detail layer for the new synthetic performance, which is in sync with the dubbed audio. Photo-realistic compositing is finally achieved by capturing a dense face albedo map of the actor's face and synthesizing a plausible mouth interior via a 3D teeth proxy and image warping. This approach shows results of superior quality when compared to image-based compositing (Chapter 4) that suffer from bleeding and ghosting artifacts.

Motivated by the inability to track performances in arbitrary videos as well as estimate and parametrize person-specific mid-scale deformations, Chapter 7 proposes a fully-parametric personalized face capture method that inverts the image formation process to reconstruct face models with multiple layers of details from unconstrained footage, e. g., videos downloaded from the Internet. The heart of this approach is a novel parametric face prior that jointly encodes the camera model as well as plausible appearance and shape changes. The appearance is modeled by skin albedo and scene lighting, whereas the shape is encoded by a subspace of coarse facial identity and expressions, person-specific medium-scale correctives, and fine-scale skin details. These layers and other related parameters are optimized automatically in a common inverse rendering framework.

Chapter 8 goes beyond face tracking and presents an automatic data-driven approach to the creation of detailed, personalized 3D face rigs from arbitrary monocular performance capture data. The reconstructed face rigs are based on three distinct shape layers (Chapter 7) and learned by coupling the coarse layer to the medium- and the fine-scale detail layer through a sparse linear regression approach. Such coupling allows us to conveniently drive the rigs with intuitive blendshape controllers to easily perform video editing and animation tasks.

Finally, Chapter 9 presents an effective data-driven approach for the automatic reconstruction of detailed and expressive 3D lip shapes, along with the dense geometry of the entire face, from monocular 2D video. Accurate lip shape is learned from a new database of high-quality multi-view reconstructions using a robust gradient-domain non-linear regressor. The proposed regressor is trained to infer accurate lip shapes from suboptimal monocular reconstructions and automatically detected inner and outer 2D lip contours. Current results demonstrate superior quality when compared to state-of-the-art face capture (Chapter 7), especially for challenging lip motions.

Discussion As stated in Chapter 1, capturing detailed facial models is the key to the success of the digitization pipeline and any inaccuracies adversely affect the quality of the animation and editing step. In arbitrary monocular setups, reconstructing 3D models is per se an ill-posed problem, since there is no information about depth and scene illumination. The presence of partial occlusions, extreme head rotations, and lighting changes may render 3D face digitization even more difficult and cause reconstruction artifacts. The state-of-the-art advances proposed in this thesis (see Chapter 5 and Chapter 7) now allow us to estimate the incident lighting and track 3D models even for challenging head rotations and expressive facial motions at high accuracy, thus assuring sufficient realism in the animation and editing of digital faces. Furthermore, advances in lip tracking (see Chapter 9) now enable us to handle strong deformations and disocclusions of the lips, which are of paramount relevance in speech-related applications and photo-realistic facial animation. Remaining challenges, such as harsh occlusions and local lighting changes, were not covered in this thesis and are further discussed in Section 10.3.



(a) (b) (c)

Figure 10.2: Limitations in the capture and animation of face rigs. (a) The eyes are not modeled but rendered as a planar 3D surface with static albedo. (b) No tongue model is reconstructed. (c) The upper and lower tooth rows are modeled with generic teeth proxies, which may lead to unrealistic 3D animations for extreme poses.

Currently, the proposed framework, mainly the face capture step, runs offline. As stated in Chapter 1, this thesis aims to obtain high-quality results with no (or minimal) user interaction, and *not* to achieve realtime performance. The capture and animation of 3D face models from a few minutes of 2D footage take several hours, which together is a drastic improvement compared to weeks of manual work. However, immediate feedback is always a desirable feature and could help detect any potential mistakes at early stages of the digitization process. Follow-up work in realtime face capture has been conducted in this regard and is discussed in Section 10.2.1.

Advances presented in Chapter 8 enable the creation and animation of detailed and fully-controllable 3D face rigs from standard 2D video input. The reconstructed rigs, however, do not model the entire human head and therefore lack important facial features, for instance, the tongue, the teeth, and scalp hair. In Chapter 6, we have proposed a simple solution based on a generic 3D teeth proxy with the appearance of the actor's teeth to model part of the oral cavity. While such an approach produces plausible video compositing results in speech-related applications, it fails to produce realistic facial animations under extreme facial expressions and non-frontal poses, as seen in Figure 10.2 (c). Reconstructing realistic 3D teeth has been recently addressed in a joint work and is briefly reviewed in Section 10.2.2. The reconstruction of other parts of the head, e. g., the tongue, remains an open scientific question that will be examined in Section 10.3.2.

10.2 Extensions

This section briefly describes two relevant follow-up works concerning the challenges that were not addressed in this thesis.

10.2.1 Realtime Performance Capture

State-of-the-art approaches, which solve a similar non-linear optimization problem to that presented in Chapters 5 and 7, have demonstrated that a drastic reduction in computing time is feasible by harnessing the data parallel processing power of the GPU [Thies et al. 2016; Wu et al. 2014; Zollhöfer et al. 2014].

A joint project, which aims at realtime face capture *in the wild*, has been recently conducted with

Figure 10.3: Reconstructed 3D teeth (right) from a set of photographs (left). The synthetic teeth accurately fit the input data when overlaid over the images (center).

other members of the Graphics, Vision, and Video (GVV) group. In this project, we have implemented the multilayer method introduced in Chapter 7, particularly the optimization of the coarse and medium layer, on the GPU using the CUDA multi-threading model in a similar way to [Thies et al. 2016]. However, we perform the optimization of the 3D face surface at the vertex level.

To facilitate data parallelization, we replaced the Levenberg-Marquardt optimization stage described in Section 7.6 by Gauss-Newton. At every step of the Gauss-Newton algorithm, an optimal linear update $\hat{\delta}$ is obtained by solving the normal equations $J^\top J\delta = -J^\top F$ via preconditioned conjugate gradient, where J is the Jacobian matrix and F denotes the vector of residuals of the objective function. Both $J^\top J$ and $-J^\top F$ are efficiently computed on the GPU. Note that the parameters of the different layers are now jointly optimized, as opposed to the multi-step optimization strategy defined in Section 7.6. Such a data parallelization strategy now allows us to achieve realtime tracking of the coarse and medium layer. We also expect a drastic reduction in the computation of the fine layer in the future. We equally foresee that the algorithms proposed for facial animation and face editing will benefit from the data parallelization on the GPU.

10.2.2 Beyond Face Capture: Model-based Teeth Reconstruction

The reconstruction of detailed teeth models has not received much attention but is crucial for the digitization of avatars that must produce realistic expressions. In a recent co-authored paper published at Siggraph Asia [Wu et al. 2016], we have presented a model-based, lightweight approach for non-invasive reconstruction of person-specific high-quality tooth rows, and also gums, from a short monocular video clip or a sparse set of photographs (see Figure 10.3). The reconstructed models can not only be used for digital actors, but also in medical applications for quick prototyping.

The key component of the proposed approach is a novel parametric tooth row prior for the upper and lower teeth that is learned from a database of 86 high-quality dental scans. The prior encodes the local shape variation of each tooth relative to an average tooth shape, the pose variation of each tooth within the tooth row, as well as the global position and scale of the entire tooth row. Plausible pose variation of each tooth is modeled as a multivariate Gaussian distribution and learned from the database. Local shape variation is modeled using principal component analysis (PCA).

We also contribute a novel fitting approach that leverages the prior mentioned above as well as automatically detected teeth contours to fit the tooth rows to the visible teeth regions while still synthesizing plausible geometry for occluded teeth. Since teeth boundaries are not fixed, the optimization is implemented in an expectation-maximization framework. To account for shapes and poses not explained by the prior, out-of-space deformation via Laplacian regularization is performed in a second step. The deformed 3D model is colored via projective texturing and smoothly blended in with the default colors of the model in occluded regions to obtain photo-realistic appearance for

the teeth and gums (see Figure 10.3).

Note that this is the first method to reconstruct a personalized teeth and gum model at high fidelity using a lightweight capture setup, e. g., a handheld device. These advances take us another step closer to the digitization of photo-realistic 3D head avatars, though there are still limitations: The teeth contours must be accurately detected to obtain high-quality 3D reconstructions. This is not always guaranteed for unconstrained input containing shadows and appearance changes in the mouth interior. Additional constraints, such as inter-teeth collision and shape-from-specularity constraints, could improve robustness and fitting quality. Furthermore, manual initialization is required to ensure convergence. Such a problem could be alleviated by anchoring the teeth to the tracked face model. In fact, combining constraints of both models show great potential for accurate face tracking, and we hope this will inspire future work.

10.3 Future Work and Outlook

In this section, we discuss other remaining aspects not covered in this thesis, including challenges in the capture of the outer face and head geometry, as well as the reconstruction of other facial features, e. g., tongue, eyes, and hair. We also share some thoughts on prospective research directions.

10.3.1 Challenges in Face Capture

In this thesis, we have assumed a pure Lambertian reflectance model to make face reconstruction a mathematically tractable problem. However, the human face, especially the skin, is a complex object that scatters and reflects incident light. As a consequence, the Lambertian assumption introduces erroneous shape details in the presence of specular highlights and can also produce oversmoothed face surfaces during rendering, as illustrated in Figure 10.4. Furthermore, extreme lighting, e. g., directional spotlights, can also lead to artifacts.

Given the problems mentioned above, a natural follow-up research topic is to decompose the face in its intrinsic image components, i. e., diffuse shading, specular highlights, subsurface scattering of skin, and possibly also albedo changes, as proposed in [Li et al. 2014; Li et al. 2015a]. Such a decomposition could contribute not only to improve the tracking and the reconstruction of detailed face models but also to edit video-realistic digital faces with advanced effects, e. g., makeup [Li et al. 2015a]. Intrinsic face decomposition in video, however, is quite ambitious because temporal consistency must be preserved, even in the presence of fast motion. Besides, robust, yet efficient, parametrization of the intrinsic layers must also be guaranteed to reconstruct photo-realistic face appearance models on videos of different length in a reasonable time. In this respect, alternative parametrizations to conventional expensive physically-based models could be explored for efficiency. Some options may include the use of wavelet bases to represent specular reflections [Li et al. 2013a] and texture space diffusion to simulate scattering of light in the skin [Borshukov and Lewis 2003], particularly during rendering.

Another important assumption made in this thesis is that the face surface is not occluded by external objects. The presence of occluding objects are known to cause artifacts in the generation of the personalized albedo map as well as the reconstruction of fine-scale skin details, e. g., wrinkles. The tracking algorithms proposed in Chapter 7, however, are still robust to mild occlusions, such as hair on the forehead and light beards, thanks to the use of dense photometric correspondences. Strong occlusions, e. g., hands in front of the face, pose a major problem for the tracking of the coarse- and

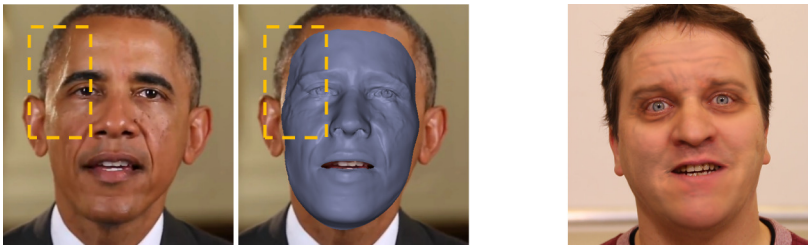


Figure 10.4: Limitations of the Lambertian reflectance assumption in the reconstruction and synthesis of faces. *Left to right:* Reconstruction artifacts due to specular highlights on the face. Renderings with over-smoothed facial details.

the medium-scale layer, since these steps rely on landmark detection algorithms and priors that can only understand image cues having distinctive facial features.

Very recently, Saito et al. [2016] employed deep learning to explicitly segment the face image from occluding objects (e. g., hands, hair, glasses, and shadows) and showed that a clear foreground and background separation could drastically help increase the robustness of tracking in 2D videos. Even though they mainly show the potential of the segmentation for regressing coarse 3D shape, such a segmentation could additionally be employed to discard photometric constraints in non-facial regions to allow a reconstruction of fine-scale surface details that is free of artifacts. There are still some important aspects to be considered. As the face segmentation works on a per-frame basis, flickering boundaries appear when generating the segmentation masks, especially when occluding objects suddenly (dis)appear, resulting in high-frequency jitter during 3D shape estimation. As such, temporal priors should additionally be incorporated to enforce temporal robustness during tracking. Still, we can anticipate that further advances in face segmentation will open up new directions in the capture of 3D digital faces from *in the wild* setups.

10.3.2 Beyond Face Capture: Tongue, Eyes, and Hair Reconstruction

The capture framework proposed in this thesis shares the limitation of related work that no detailed eye(lid) and tongue models can be reconstructed from a monocular video alone (see Figure 10.2). Possible approaches to tackle this problem are explored in the following.

The synthesis of photo-realistic tongues is a desired feature in facial animation that can help improve not only realism but also speech comprehension. Lighting changes and recurrent occlusions in the oral cavity, however, make the reconstruction of tongues from visual cues alone difficult to achieve. As such, tongue models are typically synthesized from speech using a large corpus of high-resolution 3D faces that are correlated to audio data [Anderson et al. 2013a; Ypsilos et al. 2004]. Kawai et al. [2015] alternatively leveraged a large image set of 2D tongue and mouth deformations connected to audio to render quasi 3D animations of the mouth interior during speech. These techniques, however, mainly focus on plausible speech animations, not on the synthesis of photo-realistic 3D tongue models. A recent work proposed by Hewer et al. [2014; 2016] can now deform a template mesh to MRI scans of the tongue muscle and learn a multilinear model that encodes the shape and motion of the tongue extracted from multiple persons during speech. This opens up a world of opportunities in mouth capture from video. In this thesis, we have already learned that model-based approaches with strong priors offer a good trade-off between reconstruction quality and robustness when solving ill-posed problems like tongue reconstruction. As the tongue is mostly non-visible, however, additional visual constraints in the lip and jaw region could be needed

to estimate the motion of the tongue. The correlation of the tongue motion to audio units, e. g., triphones [Bregler et al. 1997] or variable-length phonemes [Ma et al. 2006; Taylor et al. 2012], could also increase accuracy and help capture coarticulation effects. Still, several challenges remain. The appearance of the tongue is hard to capture due to occlusions and shadows, but it could be acquired in advance from HD images of the oral cavity. Handling collisions with the lips and the teeth is also a non-trivial task.

In this thesis, the eye region is represented as a simple 3D planar surface with a static eye albedo map, which is then rendered under the estimated illumination (or any user-defined lighting conditions), as shown in Figure 10.2 (a). Specially scanned eye models [Bérard et al. 2014] or synthetic eyeball templates [Ichim et al. 2015] could be textured with an estimate or the actual color of the actor's eyes to produce photo-realistic animations of the human eye. To improve realism, the movement of the eyes can be controlled by tracking the gaze of the actor in video, as proposed in [Wang et al. 2016]. The reconstruction of a detailed, actor-specific eye model from a single video is still very ambitious due to fast head motion, partial occlusions, and person-specific appearance patterns in the sclera and the iris. Bérard et al. [2016] have recently shown that photo-realistic, personalized eye models can be accurately captured from single images by leveraging a database of high-quality pre-captured eye scans. This database allows for the creation of a statistical prior that models detailed shape and appearance and that can be fitted to images using manual annotations. So far, this is just the first step towards automatic digitization of eyes in unconstrained setups. It can be expected that when combined with recent advances in eye-gaze tracking [Wang et al. 2016] photo-realistic eye capture will be possible in arbitrary 2D video footage.

Another relevant aspect beyond the scope of this thesis is the reconstruction of hair models, which is non-trivial due to the convoluted structures of the hair. Recent advances in the area now allow the reconstruction of static 3D hair models from single images. Here, we can find generative approaches that combine shape-from-shading based refinement with helical 3D priors to reconstruct fine-grained hair models [Chai et al. 2015]. We can also find hybrid methods that first search and combine candidate exemplars from a large 3D database and then optimally deform the combined model based on dense correspondences and estimated orientations of the hair strands [Chai et al. 2016; Hu et al. 2015]. In principle, such approaches could be seamlessly integrated into the proposed face capture framework to add a static hair layer to the 3D face models. Such a hair layer could be obtained either from a single image or by averaging over multiple frames. While this strategy is feasible and sufficient for short hair styles, the reconstruction of long hair styles in unconstrained video may pose a problem as it is view dependent and lacks physics-based priors. As a result, current approaches may fail to capture temporally-coherent hair dynamics and learn hair strand motion that correlates with the head pose. Alternatively, dynamics could be simulated with ad-hoc physics models during animation [Chai et al. 2014] or approximated by interpolating between static reconstructions [Cao et al. 2016]. Learning possible deformations from video is also an interesting future avenue that could replace complex modeling and physically-based simulations.

The interest in modeling and capturing hair, eyes, and tongues from optical data is quickly growing and drawing attention to the scientific community. We can anticipate that hybrid approaches that fully exploit discriminative and generative (prior) models from high-quality examples will be the key to the reconstruction of robust, accurate, and detailed head models from unconstrained video setups.

10.4 Closing Remarks

The work presented in this thesis has been motivated by current limitations in the digitization pipeline, i. e., restrictive capture and manually extensive work, and also by the inability of lightweight approaches to create photo-realistic virtual faces in unconstrained setups. Several scientific contributions, which significantly advance the state of the art in monocular facial performance capture and face capture-based video editing, have been proposed in Chapters 4–9 to deal with the limitations mentioned above. In fact, we have improved the toolbox available for creating photo-realistic human face avatars from unconstrained 2D video footage.

Results attained on challenging application scenarios have confirmed the scientific advances in the field and have shown great potential to automatize the digitization process. Animation artists can now utilize the results obtained by our algorithms as high-quality prototypes to sketch facial animations and editing effects without going through the entire conventional digitization process in post-production, thus saving money and weeks of strenuous effort. Advances in the field also help democratize the digitization technology.

Automatic digitization of photo-realistic human faces from monocular video has also recently drawn the attention of other researchers in the field. Interesting follow-up work has been carried out towards full head avatar digitization [Bérard et al. 2016; Cao et al. 2016; Ichim et al. 2015; Wu et al. 2016]. There are still many challenges that need to be solved to digitize characters anywhere and everywhere at high fidelity, as discussed in Section 10.3. We still believe that the proposed scientific contributions have paved the way for a new generation of lightweight, automatic techniques for capture, animation, and editing in movies and games. Contacts from VFX studios of Technicolor (MPC and The Mill) have confirmed the importance of our contributions and have already expressed great interest in this work.

We hope that this thesis motivates the development of more sophisticated methods, e. g., the fusion of generative and discriminative models, to digitize photo-realistic virtual models of entire heads of a quality comparable to the standard pipeline in post-production, even for *in the wild* monocular setups.

Appendices

Appendix A

Multilayer Model-based Face Capture in Unconstrained Setups

A.1 Test Sequences: Description and Specifications

The approach presented in Chapter 7 was evaluated on 9 different sequences, shown in Figure A.1. They consist of five videos (SUBJECT1[1], SUBJECT2[2], SUBJECT3[3], SUBJECT4[4], SUBJECT5[5]) captured indoors and outdoors under unknown and general lighting, and four legacy videos (ARNOLD YOUNG[6], ARNOLD OLD[7], OBAMA[8], BRYAN[9]) freely available on the Internet and downloaded from YouTube. Further descriptions of the videos and more specifications are provided below.

SUBJECT1 Studio sequence captured indoors and employed in [Valgaerts et al. 2012b]. A stereo reconstruction of this sequence is available on the Internet. The sequence consists of 714 frames with a resolution of 1088×1920 pixels. The images were downsampled to half their resolution to track the coarse and medium layer, but all other steps use full resolution images.

SUBJECT2 Studio sequence captured indoors and used in Chapters 5–6. An audio channel is also available. The complete sequence consists of 2000 frames with a resolution of 1088×1920 pixels. The images were downsampled to half their resolution to track the coarse and medium layer, but all other steps use full resolution images.

[1] http://gvv.mpi-inf.mpg.de/projects/FaceCap/

[2] http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/

[3] http://graphics.ethz.ch/publications/papers/paperBee11.php

[4] http://gvv.mpi-inf.mpg.de/projects/MonFaceCap/

[5] http://www.disneyresearch.com/project/facial-performance-enhancement/

[6] https://youtu.be/BkX2CMCXhM8

[7] https://youtu.be/EgvdhvKreJI

[8] https://youtu.be/d-VaUaTF3_k

[9] http://students.cse.tamu.edu/fuhaoshi/FacefromVideo/index.htm

Figure A.1: Test Sequences. *Top row (from left to right):* ARNOLD YOUNG, ARNOLD OLD, OBAMA, and BRYAN. *Bottom row (from left to right):* SUBJECT1, SUBJECT2, SUBJECT3, SUBJECT4, and SUBJECT5.

SUBJECT3 Another studio sequence captured indoors and employed in [Beeler et al. 2011]. Note that the actual capture setup consists of 6 high-quality cameras, one recording the actor from a frontal view. The sequence consists of 347 frames with a resolution of 864×1174 pixels. The images were downsampled to half their resolution to track the coarse and medium layer, but all other steps use full resolution images.

SUBJECT4 Outdoor sequence employed in Chapter 5 (and also in [Shi et al. 2014]). Here the actor was recorded with a GoPro Hero 3 camera from a frontal view; however, the sequence also shows challenging out-of-plane head rotations. This sequence consists of 651 frames at full HD resolution (i. e., 1920×1080 pixels). The images were downsampled to half their resolution to track the coarse and medium layer, but all other steps use full resolution images.

SUBJECT5 This sequence shows a cluttered scene captured outdoors with an iPhone camera and it was employed in [Bermano et al. 2014]. It consists of 806 frames at full HD resolution (i. e., 1920×1080 pixels). The images were downsampled to half their resolution to track the coarse and medium layer, but all other steps use full resolution images.

ARNOLD YOUNG This video shows an interview with Arnold Schwarzenegger about the launch of the movie "Predator". The sequence consists of a subset of 1489 frames with a resolution of 480×360 pixels. The video was processed at its original full resolution in all steps of the pipeline.

ARNOLD OLD This video shows Arnold Schwarzenegger's message for DECC's Energy Efficiency Mission Launch. The sequence consists of a subset of 1000 frames with a resolution of 1280×720 pixels. The video was processed at its original full resolution in all steps of the pipeline.

OBAMA This video shows a greeting address by president Obama who commemorates Independence Day on July 4. The sequence consists of a subset of 961 frames with a resolution of

1280×720 pixels. The video was processed at its original full resolution in all steps of the pipeline.

BRYAN This video shows the actor Bryan Lee Cranston talking about the end of his journey with the TV series "Breaking Bad". The pipeline was run on a subset of 702 frames at a resolution of 640×360 pixels.

A.2 Energy Function: Derivatives

A.2.1 Data Objective

Feature Term Let us first rewrite Equation 7.13, as follows:

$$E_{feature}(\mathcal{X}) = \sum_{\ell=1}^{L} \left(f \frac{\bar{\mathbf{v}}_{x,n_\ell}}{\bar{\mathbf{v}}_{z,n_\ell}} + c_x - \mathbf{y}_{x,\ell} \right)^2 + \sum_{\ell=1}^{L} \left(f \frac{\bar{\mathbf{v}}_{y,n_\ell}}{\bar{\mathbf{v}}_{z,n_\ell}} + c_y - \mathbf{y}_{y,\ell} \right)^2 \ , \tag{A.1}$$

where f denotes the focal length, $c \in \mathbb{R}^2$ is the principal point and $\bar{\mathbf{v}} = \mathbf{R}\hat{\mathbf{v}} + \mathbf{t}$. Note that $\mathbf{R}$ and $\mathbf{t}$ denote the rotation matrix and translation vector, respectively.

Let $\phi_{x,n_\ell} = f \frac{\bar{\mathbf{v}}_{x,n_\ell}}{\bar{\mathbf{v}}_{z,n_\ell}} + c_x$, $\phi_{y,n_\ell} = f \frac{\bar{\mathbf{v}}_{y,n_\ell}}{\bar{\mathbf{v}}_{z,n_\ell}} + c_y$, $\forall \ell$, and $a = \{x, y, z\}$ be one axis of the Cartesian coordinate system. The derivatives with respect to α, δ and τ then read as follows:

1.

$$\begin{aligned} \frac{\partial E_{feature}(\mathcal{X})}{\partial \alpha_k} = {} & 2f \cdot \sum_{\ell=1}^{L} (\phi_{x,n_\ell} - \mathbf{y}_{x,\ell}) \cdot \left(\frac{\partial \bar{\mathbf{v}}_{x,n_\ell}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{z,n_\ell} - \frac{\partial \bar{\mathbf{v}}_{z,n_\ell}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{x,n_\ell} \right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n_\ell}^2} \\ & + 2f \cdot \sum_{\ell=1}^{L} (\phi_{y,n_\ell} - \mathbf{y}_{y,\ell}) \cdot \left(\frac{\partial \bar{\mathbf{v}}_{y,n_\ell}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{z,n_\ell} - \frac{\partial \bar{\mathbf{v}}_{z,n_\ell}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{y,n_\ell} \right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n_\ell}^2} \ , \ \forall k \in (1, K_s) \ , \end{aligned} \tag{A.2}$$

where $\frac{\partial \bar{\mathbf{v}}_{n_\ell}}{\partial \alpha_k} = \mathbf{R}\hat{\mathbf{E}}^s_{n_\ell,k}$ and $\hat{\mathbf{E}}^s = \mathbf{E}_s \Sigma_s$.

Note that $\hat{\mathbf{E}}^s_{n_\ell,k} \in \mathbb{R}^3$ is the vector corresponding to the k-th column of $\hat{\mathbf{E}}^s$ at vertex index n_ℓ.

2.

$$\begin{aligned} \frac{\partial E_{feature}(\mathcal{X})}{\partial \delta_k} = {} & 2f \cdot \sum_{\ell=1}^{L} (\phi_{x,n_\ell} - \mathbf{y}_{x,\ell}) \cdot \left(\frac{\partial \bar{\mathbf{v}}_{x,n_\ell}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{z,n_\ell} - \frac{\partial \bar{\mathbf{v}}_{z,n_\ell}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{x,n_\ell} \right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n_\ell}^2} \\ & + 2f \cdot \sum_{\ell=1}^{L} (\phi_{y,n_\ell} - \mathbf{y}_{y,\ell}) \cdot \left(\frac{\partial \bar{\mathbf{v}}_{y,n_\ell}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{z,n_\ell} - \frac{\partial \bar{\mathbf{v}}_{z,n_\ell}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{y,n_\ell} \right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n_\ell}^2} \ , \ \forall k \in (1, K_e) \ , \end{aligned} \tag{A.3}$$

where $\frac{\partial \bar{\mathbf{v}}_{n_\ell}}{\partial \delta_k} = \mathbf{R}\hat{\mathbf{E}}^e_{n_\ell,k}$ and $\hat{\mathbf{E}}^e = \mathbf{E}_e \Sigma_e$.

Note that $\hat{\mathbf{E}}^e_{n_\ell,k} \in \mathbb{R}^3$ is the vector corresponding to the k-th column of $\hat{\mathbf{E}}^e$ at vertex index n_ℓ.

3.

$$\frac{\partial E_{feature}(\mathcal{X})}{\partial \tau_{a,k}} = 2f \cdot \sum_{\ell=1}^{L} (\phi_{x,n_\ell} - \mathbf{y}_{x,\ell}) \cdot \left(\frac{\partial \bar{\mathbf{v}}_{x,n_\ell}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{z,n_\ell} - \frac{\partial \bar{\mathbf{v}}_{z,n_\ell}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{x,n_\ell} \right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n_\ell}^2}$$
$$+ 2f \cdot \sum_{\ell=1}^{L} (\phi_{y,n_\ell} - \mathbf{y}_{y,\ell}) \cdot \left(\frac{\partial \bar{\mathbf{v}}_{y,n_\ell}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{z,n_\ell} - \frac{\partial \bar{\mathbf{v}}_{z,n_\ell}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{y,n_\ell} \right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n_\ell}^2} \ ,$$
$$\forall a = \{x,y,z\}, \ \ \forall k \in (1, K_c) \ , \tag{A.4}$$

where $\frac{\partial \bar{\mathbf{v}}_{n_\ell}}{\partial \tau_{a,k}} = \mathbf{R} \mathbb{1}_a H_{n,k}$.

Here $H_{n,k} \in \mathbb{R}$ is the n-th element of H_k and $\mathbb{1}_a \in \mathbb{R}^3$, $\forall a = \{x,y,z\}$ are mutually orthogonal unit vectors corresponding to the x-, y- or z-axis of the Cartesian coordinate system (e. g., $\mathbb{1}_x = [1,0,0]^\top$), and $\tau_{a,k}$ is the k-th deformation coefficient that parametrizes deformations in the a-axis.

Photo-consistency Term Let us first rewrite Equation 7.12, as follows:

$$E_{photo}(\mathcal{X}) = \sum_{n=1}^{N} \| f_t[\phi_n] - \mathcal{B}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma) \|_2^2 \tag{A.5}$$
$$= \sum_{n=1}^{N} \sum_{\omega=1}^{3} \left(f_t^\omega[\phi_n] - \mathcal{B}^\omega(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma) \right)^2 \tag{A.6}$$
$$= \sum_{n=1}^{N} \sum_{\omega=1}^{3} \left(f_t^\omega[\phi_n] - \mathbf{c}_n^\omega \cdot \sum_{b=1}^{B^2} \gamma_b^\omega \cdot \mathrm{Y}_b(\hat{\mathbf{n}}_n) \right)^2 \ , \tag{A.7}$$

where $\mathbf{c}_n^\omega$ is the skin albedo at vertex $\hat{\mathbf{n}}_n$ corresponding to the ω-th color channel. $\mathrm{Y}_b(\hat{\mathbf{n}}_n)$ and γ_b, $\forall b$ denote the spherical harmonics (SH) functions (parametrized in terms of $\hat{\mathbf{n}}_n$) and their corresponding weights, respectively. The SH functions are shown in Table A.1. Note that $\phi_{x,n} = f\frac{\bar{\mathbf{v}}_{x,n}}{\bar{\mathbf{v}}_{z,n}} + c_x$ and $\phi_{y,n} = f\frac{\bar{\mathbf{v}}_{y,n}}{\bar{\mathbf{v}}_{z,n}} + c_y$ are the x and y coordinates of vertex $\bar{\mathbf{v}}_n$ projected onto the image plane.

Let us define $\tilde{\mathbf{n}}_n = \sum_{h=1}^{|\mathcal{A}|} (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}_h^1) \times (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}_h^2)$ and $d_n = \| \tilde{\mathbf{n}}_n \|_2$, $\forall n$ as the non-normalized normal at vertex $\bar{\mathbf{v}}_n$ and its respective normalization factor, where $\mathcal{A}$ denotes the set of the triangle faces adjacent to $\bar{\mathbf{v}}_n$, and $\bar{\mathbf{v}}_h^1$, $\bar{\mathbf{v}}_h^2$ are the two vertices of the h-th triangle face adjacent to vertex $\bar{\mathbf{v}}_n$.

The derivatives with respect to β, γ, α, δ and τ then read as follows:

1.

$$\frac{E_{photo}(\mathcal{X})}{\partial \beta_k} = -2 \sum_{n=1}^{N} (f_t[\phi_n] - \mathcal{B}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \circ \hat{\mathbf{E}}_{n,k} \circ \hat{L}_n \ , \ \forall k \tag{A.8}$$
$$= -2 \sum_{n=1}^{N} \sum_{\omega=1}^{3} (f_t^\omega[\phi_n] - \mathcal{B}^\omega(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \hat{\mathbf{E}}_{n,k}^\omega \cdot \hat{L}_n^\omega \ , \ \forall k \in (1, K_r) \ , \tag{A.9}$$

where $\hat{L}_n = \sum_{b=1}^{B^2} \gamma_b \cdot \mathrm{Y}_b(\hat{\mathbf{n}}_n)$, $\hat{\mathbf{E}} = \mathbf{E}_r \Sigma_r$ and $\circ$ denotes a point-wise multiplication. Note that $\hat{\mathbf{E}}_{n,k} \in \mathbb{R}^3$ is a RGB vector corresponding to the k-th column of $\hat{\mathbf{E}}$ at vertex index n.

Table A.1: Spherical harmonics functions $\mathrm{Y}_b(\hat{\mathbf{n}})$ and their derivatives $\frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}})}{\partial e}$, both parametrized in terms of the vertex normal $\hat{\mathbf{n}}$. Note that $\partial e, \forall e = \{\alpha, \delta, \tau\}$ are the partial derivatives w. r. t. the shape, expression and corrective weights.

b	$\mathrm{Y}_b(\hat{\mathbf{n}})$	$\frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}})}{\partial e}$
1	$\sqrt{\frac{1}{4\pi}}$	0
2	$\sqrt{\frac{3}{4\pi}}\hat{\mathbf{n}}_y$	$\sqrt{\frac{3}{4\pi}}\frac{\partial \hat{\mathbf{n}}_y}{\partial e}$
3	$\sqrt{\frac{3}{4\pi}}\hat{\mathbf{n}}_z$	$\sqrt{\frac{3}{4\pi}}\frac{\partial \hat{\mathbf{n}}_z}{\partial e}$
4	$\sqrt{\frac{3}{4\pi}}\hat{\mathbf{n}}_x$	$\sqrt{\frac{3}{4\pi}}\frac{\partial \hat{\mathbf{n}}_x}{\partial e}$
5	$\sqrt{\frac{15}{16\pi}}(\hat{\mathbf{n}}_x^2 - \hat{\mathbf{n}}_y^2)$	$2\sqrt{\frac{15}{16\pi}}(\hat{\mathbf{n}}_x\frac{\partial \hat{\mathbf{n}}_x}{\partial e} - \hat{\mathbf{n}}_y\frac{\partial \hat{\mathbf{n}}_y}{\partial e})$
6	$\sqrt{\frac{15}{4\pi}}\hat{\mathbf{n}}_z\hat{\mathbf{n}}_x$	$\sqrt{\frac{15}{4\pi}}(\hat{\mathbf{n}}_z\frac{\partial \hat{\mathbf{n}}_x}{\partial e} + \hat{\mathbf{n}}_x\frac{\partial \hat{\mathbf{n}}_z}{\partial e})$
7	$\sqrt{\frac{5}{16\pi}}(3\hat{\mathbf{n}}_z^2 - 1)$	$6\sqrt{\frac{5}{16\pi}}(\hat{\mathbf{n}}_z\frac{\partial \hat{\mathbf{n}}_z}{\partial e})$
8	$\sqrt{\frac{15}{4\pi}}\hat{\mathbf{n}}_y\hat{\mathbf{n}}_z$	$\sqrt{\frac{15}{4\pi}}(\hat{\mathbf{n}}_y\frac{\partial \hat{\mathbf{n}}_z}{\partial e} + \hat{\mathbf{n}}_z\frac{\partial \hat{\mathbf{n}}_y}{\partial e})$
9	$\sqrt{\frac{15}{4\pi}}\hat{\mathbf{n}}_x\hat{\mathbf{n}}_y$	$\sqrt{\frac{15}{4\pi}}(\hat{\mathbf{n}}_x\frac{\partial \hat{\mathbf{n}}_y}{\partial e} + \hat{\mathbf{n}}_y\frac{\partial \hat{\mathbf{n}}_x}{\partial e})$

2.

$$\frac{E_{photo}(\mathcal{X})}{\partial \gamma_k} = -2\sum_{n=1}^{N}(f_t[\phi_n] - \mathcal{B}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \circ \mathbf{c}_n \circ \gamma_k \cdot \mathrm{Y}_b(\hat{\mathbf{n}}_n) \;\; , \;\; \forall k \in (1, B^2) \;\; . \tag{A.10}$$

$$\frac{E_{photo}(\mathcal{X})}{\partial \gamma_k^\omega} = -2\sum_{n=1}^{N}(f_t^\omega[\phi_n] - \mathcal{B}^\omega(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \mathbf{c}_n^\omega \cdot \gamma_k^\omega \cdot \mathrm{Y}_b(\hat{\mathbf{n}}_n) \;\; , \;\; \forall k, \omega \;\; . \tag{A.11}$$

3.

$$\begin{aligned}\frac{E_{photo}(\mathcal{X})}{\partial \alpha_k} = &-2\sum_{n=1}^{N}\sum_{\omega=1}^{3}(f_t^\omega[\phi_n] - \mathcal{B}^\omega(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \mathbf{c}_n^\omega \cdot \sum_{b=1}^{B^2}\gamma_b^\omega \cdot \frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}}_n)}{\partial \alpha_k} \\ &+2\sum_{n=1}^{N}\sum_{\omega=1}^{3}(f_t^\omega[\phi_n] - \mathcal{B}^\omega(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \frac{\partial f_t^\omega[\phi_n]}{\partial \alpha_k}, \;\; \forall k \in (1, K_s) \;\; ,\end{aligned} \tag{A.12}$$

where

$$\begin{aligned}\frac{\partial f_t^\omega[\phi_n]}{\partial \alpha_k} = &f \cdot \left(\frac{\partial f_t^\omega[\phi_n]}{\partial x} \cdot \left(\frac{\partial \bar{\mathbf{v}}_{x,n}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{z,n} - \frac{\partial \bar{\mathbf{v}}_{z,n}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{x,n}\right)\right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n}^2} \\ &+ f \cdot \left(\frac{\partial f_t^\omega[\phi_n]}{\partial y} \cdot \left(\frac{\partial \bar{\mathbf{v}}_{y,n}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{z,n} - \frac{\partial \bar{\mathbf{v}}_{z,n}}{\partial \alpha_k} \cdot \bar{\mathbf{v}}_{y,n}\right)\right) \cdot \frac{1}{\bar{\mathbf{v}}_{z,n}^2} \;\; ,\end{aligned}$$

$$\frac{\partial \bar{\mathbf{v}}_n}{\partial \alpha_k} = \mathbf{R}\hat{\mathbf{E}}_{n,k}^s \;\; ,$$

$$\frac{\partial \hat{\mathbf{n}}_n}{\partial \alpha_k} = \frac{\partial \tilde{\mathbf{n}}_n}{\partial \alpha_k} \cdot \frac{1}{d_n} - \left\langle \frac{\partial \tilde{\mathbf{n}}_n}{\partial \alpha_k}, \tilde{\mathbf{n}}_n \right\rangle \cdot \frac{\tilde{\mathbf{n}}_n}{d_n^3} \;\; ,$$

$$\frac{\partial \tilde{\mathbf{n}}_n}{\partial \alpha_k} = \sum_{h=1}^{\mathcal{A}} \mathbf{R}(\hat{\mathbf{E}}_{n,k}^s - \hat{\mathbf{E}}_{h,k}^{s,1}) \times (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}_h^2) + (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}_h^1) \times \mathbf{R}(\hat{\mathbf{E}}_{n,k}^s - \hat{\mathbf{E}}_{h,k}^{s,2}) \;\; .$$

Here $\hat{\mathbf{E}}^s = \mathbf{E}_s\Sigma_s$, $\frac{\partial f_t^\omega[\phi_n]}{\partial x}$ and $\frac{\partial f_t^\omega[\phi_n]}{\partial y}$ are the image gradients computed using Sobel operators, and $\langle\cdot\rangle$ represents the dot product. The derivatives of $\frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}})}{\partial \alpha}$ are shown in Table A.1. Note

that $\hat{\mathbf{E}}^{s,1}_{h,k}, \hat{\mathbf{E}}^{s,2}_{h,k} \in \mathbb{R}^3$ are vectors corresponding to k-th column of $\hat{\mathbf{E}}^s$ whose row index coincides with that of vertex $\bar{\mathbf{v}}^1_h, \bar{\mathbf{v}}^2_h$, respectively.

4.

$$\frac{E_{photo}(\mathcal{X})}{\partial \delta_k} = -2 \sum_{n=1}^{N} \sum_{\omega=1}^{3} (f_t^{\omega}[\phi_n] - \mathcal{B}^{\omega}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \mathbf{c}_n^{\omega} \cdot \sum_{b=1}^{B^2} \gamma_b^{\omega} \cdot \frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}}_n)}{\partial \delta_k} + 2 \sum_{n=1}^{N} \sum_{\omega=1}^{3} (f_t^{\omega}[\phi_n] - \mathcal{B}^{\omega}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \frac{\partial f_t^{\omega}[\phi_n]}{\partial \delta_k}, \quad \forall k \in (1, K_e) \ , \tag{A.13}$$

where

$$\frac{\partial f_t^{\omega}[\phi_n]}{\partial \delta_k} = f \cdot \left(\frac{\partial f_t^{\omega}[\phi_n]}{\partial x} \cdot \left(\frac{\partial \bar{\mathbf{v}}_{x,n}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{z,n} - \frac{\partial \bar{\mathbf{v}}_{z,n}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{x,n} \right) \right) \cdot \frac{1}{\bar{\mathbf{v}}^2_{z,n}} + f \cdot \left(\frac{\partial f_t^{\omega}[\phi_n]}{\partial y} \cdot \left(\frac{\partial \bar{\mathbf{v}}_{y,n}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{z,n} - \frac{\partial \bar{\mathbf{v}}_{z,n}}{\partial \delta_k} \cdot \bar{\mathbf{v}}_{y,n} \right) \right) \cdot \frac{1}{\bar{\mathbf{v}}^2_{z,n}} \ ,$$

$$\frac{\partial \bar{\mathbf{v}}_n}{\partial \delta_k} = \mathbf{R} \hat{\mathbf{E}}^e_{n,k} \ ,$$

$$\frac{\partial \hat{\mathbf{n}}_n}{\partial \delta_k} = \frac{\partial \tilde{\mathbf{n}}_n}{\partial \delta_k} \cdot \frac{1}{d_n} - \left\langle \frac{\partial \tilde{\mathbf{n}}_n}{\partial \delta_k}, \tilde{\mathbf{n}}_n \right\rangle \cdot \frac{\tilde{\mathbf{n}}_n}{d_n^3} \ ,$$

$$\frac{\partial \tilde{\mathbf{n}}_n}{\partial \delta_k} = \sum_{h=1}^{\mathcal{A}} \mathbf{R}(\hat{\mathbf{E}}^e_{n,k} - \hat{\mathbf{E}}^{e,1}_{h,k}) \times (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}^2_h) + (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}^1_h) \times \mathbf{R}(\hat{\mathbf{E}}^e_{n,k} - \hat{\mathbf{E}}^{e,2}_{h,k}) \ .$$

Here $\hat{\mathbf{E}}^e = \mathbf{E}_e \Sigma_e$. The derivatives of $\frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}})}{\partial \delta}$ are shown in Table A.1. Note that $\hat{\mathbf{E}}^{e,1}_{h,k}, \hat{\mathbf{E}}^{e,2}_{h,k} \in \mathbb{R}^3$ are vectors corresponding to k-th column of $\hat{\mathbf{E}}^e$ whose row index coincides with that of vertex $\bar{\mathbf{v}}^1_h, \bar{\mathbf{v}}^2_h$, respectively.

5.

$$\frac{E_{photo}(\mathcal{X})}{\partial \tau_{a,k}} = -2 \sum_{n=1}^{N} \sum_{\omega=1}^{3} (f_t^{\omega}[\phi_n] - \mathcal{B}^{\omega}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \mathbf{c}_n^{\omega} \cdot \sum_{b=1}^{B^2} \gamma_b^{\omega} \cdot \frac{\partial \mathrm{Y}_b(\hat{\mathbf{n}}_n)}{\partial \tau_{a,k}} + 2 \sum_{n=1}^{N} \sum_{\omega=1}^{3} (f_t^{\omega}[\phi_n] - \mathcal{B}^{\omega}(\hat{\mathbf{n}}_n, \mathbf{c}_n \mid \gamma)) \cdot \frac{\partial f_t^{\omega}[\phi_n]}{\partial \tau_{a,k}}, \quad \forall a \in \{x, y, z\}, k \in (1, K_c) \ , \tag{A.14}$$

where

$$\frac{\partial f_t^{\omega}[\phi_n]}{\partial \tau_{a,k}} = f \cdot \left(\frac{\partial f_t^{\omega}[\phi_n]}{\partial x} \cdot \left(\frac{\partial \bar{\mathbf{v}}_{x,n}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{z,n} - \frac{\partial \bar{\mathbf{v}}_{z,n}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{x,n} \right) \right) \cdot \frac{1}{\bar{\mathbf{v}}^2_{z,n}} + f \cdot \left(\frac{\partial f_t^{\omega}[\phi_n]}{\partial y} \cdot \left(\frac{\partial \bar{\mathbf{v}}_{y,n}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{z,n} - \frac{\partial \bar{\mathbf{v}}_{z,n}}{\partial \tau_{a,k}} \cdot \bar{\mathbf{v}}_{y,n} \right) \right) \cdot \frac{1}{\bar{\mathbf{v}}^2_{z,n}} \ ,$$

$$\frac{\partial \bar{\mathbf{v}}_n}{\partial \tau_{a,k}} = \mathbf{R} \mathbb{1}_a H_{n,k} \ ,$$

$$\frac{\partial \hat{\mathbf{n}}_n}{\partial \tau_{a,k}} = \frac{\partial \tilde{\mathbf{n}}_n}{\partial \tau_{a,k}} \cdot \frac{1}{d_n} - \left\langle \frac{\partial \tilde{\mathbf{n}}_n}{\partial \tau_{a,k}}, \tilde{\mathbf{n}}_n \right\rangle \cdot \frac{\tilde{\mathbf{n}}_n}{d_n^3} \ ,$$

$$\frac{\partial \tilde{\mathbf{n}}_n}{\partial \tau_{a,k}} = \sum_{h=1}^{\mathcal{A}} \mathbf{R} \mathbb{1}_a (H_{n,k} - H^1_{h,k}) \times (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}^2_h) + (\bar{\mathbf{v}}_n - \bar{\mathbf{v}}^1_h) \times \mathbf{R} \mathbb{1}_a (H_{n,k} - H^2_{h,k}) \ .$$

The derivatives of $\frac{\partial Y_b(\hat{\mathbf{n}})}{\partial \tau}$ are shown in Table A.1. Note that $H^1_{h,k}, H^2_{h,k} \in \mathbb{R}^3$ are three copies of an entry of H_k whose row index corresponds to that of vertex $\bar{\mathbf{v}}^1_h, \bar{\mathbf{v}}^2_h$, respectively.

A.2.2 Prior Objective and Boundary Constraint

Probabilistic Shape Prior: Coarse-scale Model The derivatives E_{prob1} with respect to α, β and γ are:

1.
$$\frac{\partial E_{prob1}(\alpha,\beta,\gamma)}{\partial \alpha_k} = 2w_s \frac{\alpha_k}{\sigma^2_{\alpha_k}} \ , \ \forall k \in (1, K_s) \ . \tag{A.15}$$

2.
$$\frac{\partial E_{prob1}(\alpha,\beta,\gamma)}{\partial \beta_k} = 2w_r \frac{\beta_k}{\sigma^2_{\beta_k}} \ , \ \forall k \in (1, K_r) \ . \tag{A.16}$$

3.
$$\frac{\partial E_{prob1}(\alpha,\beta,\gamma)}{\partial \gamma_k} = 2w_l \frac{\gamma_k}{\sigma^2_{\gamma_k}} \ , \ \forall k \in (1, B^2) \ . \tag{A.17}$$

To compute the derivatives of the sparsity prior term in the Levenberg-Marquardt algorithm, we redefine Equation 7.17 as follows:

$$E_{sparse}(\delta) = w_d \sum_{k=1}^{K_e} \upsilon^{(i)}_k \delta^2_k \ , \tag{A.18}$$

where $\upsilon^{(i)}_k$ is an iterative weight assigned to δ^2_k at each i-th iteration. The derivatives of E_{sparse} with respect to δ then read as follows:

1.
$$\frac{\partial E_{sparse}(\delta)}{\partial \delta_k} = 2w_d \upsilon^{(i)}_k \delta_k \ , \ \forall k \in (1, K_e) \ . \tag{A.19}$$

At the first iteration $\upsilon^{(0)}_k = 1$. In the next iterations, $\upsilon^{(i)}_k = 1/(|\delta^{(i)}_k| + 0.00001)$.

Probabilistic Shape Prior: Medium-scale Model The derivatives of E_{prob2} with respect to τ are:

1.
$$\frac{\partial E_{prob2}(\tau)}{\partial \tau_k} = 2w_z \frac{\tau_k}{\sigma^2_{\tau_k}} + 2w_t(\tau_k - \tau^{prev}_k) \ , \ \forall k \in (1, K_c) \ . \tag{A.20}$$

Boundary Constraint The derivatives of E_{bound} with respect to δ read as follows:

1.

$$\frac{\partial E_{bound}(\delta)}{\partial \delta_k} = w_b \begin{cases} 2\delta_k, & \text{if } \delta_k < 0 \\ 0, & \text{if } 0 \geq \delta_k \geq 1 \\ 2(\delta_k - 1), & \text{if } \delta_k > 1 \end{cases} , \ \forall k \in (1, K_e) \ . \tag{A.21}$$

Appendix B

Beyond Face Capture: Accurate Lip Tracking

B.1 High-quality Lip Database: Training Examples

All lip motions that were captured to train the radial basis functions network presented in Chapter 9 are given in Table B.3, Table B.1, Table B.2, and Table B.4. The motions were performed sequentially by the different subjects (S1, S2, S3, and S4), as listed in the corresponding tables (please refer to the additional supplemental video at the project website for more details [1]). To train the generalization regressor (**GR**), we stacked the captured data of three or four of the subjects (three if the test subject is one out of the four subjects) in the following order: S2, S3, S1, S4. The multiple subject regressor (**MS**) uses the same order.

[1] http://gvv.mpi-inf.mpg.de/projects/MonLipReconstruction/

Table B.1: High-quality lip database: Performed lip motions - Subject S2.

Action	Features	# Frames
Roll lips	Outwards to inwards; mouth closed	51
Roll lips	Outwards to inwards; mouth half-open	47
Roll lips	Inwards to outwards; mouth open	39
Smile	Mouth closed	17
Smile	Mouth half-open	49
Smile	Mouth open	51
Move left	Mouth closed	30
Move right	Mouth closed	32
Move left	Mouth half-open	30
Move right	Mouth half-open	26
Move left	Mouth open	28
Move right	Mouth open	21
Bite	Lower lip; mouth closed	26
Open	Mouth closed to half-open	76
Kiss	Mouth closed	33
Round lips	Mouth closed	27
Round lips	Mouth half-open to open	74
Move lip	Lower lip down; mouth closed	30
Move lip	Upper lip up; mouth closed	31
Pull corners	Mouth closed	33

Table B.2: High-quality lip database: Performed lip motions - Subject S3.

Action	Features	# Frames
Kiss	Mouth closed	25
Smile	Mouth closed	32
Smile	Mouth half-open	25
Smile	Mouth open	27
Move left	Mouth closed	18
Move left	Mouth half-open	17
Move left	Mouth open	27
Move right	Mouth closed	17
Move right	Mouth half-open	24
Move right	Mouth open	18
Pout	Mouth closed	25
Roll lips	Inwards; mouth closed	15
Roll lips	Inwards; mouth half-open	20
Roll lips	Inwards; mouth open	25
Roll lips	Outwards; mouth closed	20
Roll lips	Outwards; mouth half-open to open	76
Move lips	Both up; mouth closed	13
Move lip	Lower lip down; mouth closed	22
Move lip	Upper lip up; mouth closed	21
Pull corners	Mouth closed	21
Pull corner	Right; mouth closed	7
Pull corner	Left; mouth closed	18
Open	Mouth half-open to open	34
Open	Mouth open to wide open	30
Open	Mouth closed to half-open	42
Viseme f	Exaggerated; mouth half-open to closed	22
Viseme ʃ	Exaggerated; mouth half-open to closed	18
Viseme m	Exaggerated; mouth half-open to closed	18
Viseme θ	Exaggerated; mouth half-open to open	12
Viseme ɔː	Exaggerated; mouth half-open	13
Viseme Iː	Exaggerated; mouth half-open	14
Viseme æ	Exaggerated; mouth half-open to open	19
Viseme əʊ	Exaggerated; mouth half-open to open	16

Table B.3: High-quality lip database: Performed lip motions - Subject S1.

Action	Features	# Frames
Roll lips	Outwards; mouth half-open	48
Roll lips	Outwards; mouth open	28
Move lips	Both up; mouth closed	14
Move lip	Lower lip down; mouth closed	24
Move lip	Upper lip up; mouth closed	20
Pull corners	Mouth closed	43
Pull corner	Left; mouth closed	29
Pull corner	Right; mouth closed	19
Open	Mouth closed to half-open	34
Open	Mouth half-open to wide-open	54
Kiss	Mouth closed	37
Smile	Mouth closed	23
Smile	Mouth half-open	38
Smile	Mouth open	23
Move left	Mouth closed	31
Move left	Mouth half-open	21
Move left	Mouth open	32
Move right	Mouth closed	34
Move right	Mouth half-open	26
Move right	Mouth open	34
Pout	Mouth closed	29
Roll lips	Inwards; mouth closed	25
Roll lips	Inwards; mouth half-open	36
Roll lips	Inwards; mouth open	34
Roll lips	Outwards; mouth closed	31
Viseme θ	Exaggerated; mouth half-open	17
Viseme Iː	Exaggerated; mouth closed to half-open	18
Viseme əʊ	Exaggerated; mouth closed to open	32
Viseme f	Exaggerated; mouth closed	15
Viseme ʃ	Exaggerated; mouth closed	16
Viseme m	Exaggerated; mouth half-open to closed	32
Viseme ɔː	Exaggerated; mouth closed to half-open	21
Viseme æ	Exaggerated; mouth half-open to open	24

Table B.4: High-quality lip database: Performed lip motions - Subject S4.

Action	Features	# Frames
Kiss	Mouth closed	18
Smile	Mouth closed	22
Smile	Mouth half-open	31
Smile	Mouth open	31
Move left	Mouth closed	18
Move left	Mouth half-open	22
Move left	Mouth open	26
Move right	Mouth closed	18
Move right	Mouth half-open	31
Move right	Mouth open	24
Pout	Mouth closed	22
Roll lips	Inwards; mouth closed	28
Roll lips	Inwards; mouth half-open	25
Roll lips	Inwards; mouth open	25
Roll lips	Outwards; mouth closed	29
Roll lips	Outwards; mouth half-open	20
Roll lips	Outwards; mouth open	37
Move lips	Both up; mouth closed	24
Move lip	Lower lip down; mouth closed	20
Move lip	Upper lip up; mouth closed	17
Pull corners	Mouth closed	19
Pull corner	Left; mouth closed	19
Pull corner	Right; mouth closed	25
Open	Mouth closed to half-open	50
Open	Mouth half-open to wide-open	48
Open	Sticky lips; mouth closed to half-open	24
Viseme f	Exaggerated; mouth half-open to closed	21
Viseme ʃ	Exaggerated; mouth closed	23
Viseme m	Exaggerated; mouth half-open to closed	22
Viseme θ	Exaggerated; mouth half-open	21
Viseme ɔː	Exaggerated; mouth closed to half-open	23
Viseme Iː	Exaggerated; mouth closed	17
Viseme æ	Exaggerated; mouth half-open to open	27
Viseme əʊ	Exaggerated; mouth closed to open	18

Bibliography (Own Work)

GARRIDO, P., VALGAERTS, L., REHMSEN, O., THORMAEHLEN, T., PÉREZ, P. AND THEOBALT, C. (2014): Automatic Face Reenactment. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, 4217–4224 5, 45

GARRIDO, P., VALGAERTS, L., SARMADI, H., STEINER, I., VARANASI, K., PÉREZ, P. AND THEOBALT, C. (2015): VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track. *Comput. Graph. Forum (Proceedings of Eurographics 2015)*, 34 (2), 193–204 5, 85

GARRIDO, P., VALGAERTS, L., WU, C. AND THEOBALT, C. (2013): Reconstructing Detailed Dynamic Face Geometry from Monocular Video. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*, 32 (6), 158:1–158:10 5, 14, 27, 45, 67, 115, 123

GARRIDO, P., ZOLLHÖFER, M., CASAS, D., VALGAERTS, L., VARANASI, K., PÉREZ, P. AND THEOBALT, C. (2016a): Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Trans. Graph.* 35 (3), 28:1–28:15 6, 14, 105, 127

GARRIDO, P., ZOLLHÖFER, M., WU, C., BRADLEY, D., PÉREZ, P., BEELER, T. AND THEOBALT, C. (2016b): Corrective 3D Reconstruction of Lips from Monocular Video. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2016)* 6, 141

WU, C., BRADLEY, D., GARRIDO, P., ZOLLHÖFER, M., THEOBALT, C., GROSS, M. AND BEELER, T. (2016): Model-Based Teeth Reconstruction. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2016)* 165, 169

Bibliography

AGUDO, A., AGAPITO, L., CALVO, B. AND MONTIEL, J. M. M. (2014): Good Vibrations: A Modal Analysis Approach for Sequential Non-rigid Structure from Motion. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, 1558–1565 24

AHONEN, T., HADID, A. AND PIETIKAINEN, M. (2006): Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (12), 2037–2041 49

ALEXA, M. (2002): Linear Combination of Transformations. *ACM Trans. Graph.* 21 (3), 380–387 112

ALEXANDER, O., FYFFE, G., BUSCH, J., YU, X., ICHIKARI, R., JONES, A., DEBEVEC, P., JIMENEZ, J., DANVOYE, E., ANTIONAZZI, B., EHELER, M., KYSELA, Z. AND PAHLEN, J. VON DER (2013): Digital Ira: Creating a Real-time Photoreal Digital Actor. In *ACM SIGGRAPH 2013 Posters*, SIGGRAPH '13, 1:1–1:1 1, 21, 33

ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, J.-Y., MA, W.-C., WANG, C.-C. AND DEBEVEC, P. (2010): The Digital Emily Project: Achieving a Photorealistic Digital Actor. *IEEE Computer Graphics and Applications*, 30 (4), 20–31 1, 20, 21, 40, 68, 106, 110, 129

ANDERSON, R., STENGER, B. AND CIPOLLA, R. (2013a): Lip Tracking for 3D Face Registration. In *Proceedings of the 13. IAPR International Conference on Machine Vision Applications*, MVA '13, 145–148 31, 104, 157, 167

ANDERSON, R., STENGER, B., WAN, V. AND CIPOLLA, R. (2013b): Expressive Visual Text-to-Speech Using Active Appearance Models. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, 3382–3389 36

ARUN, K. S., HUANG, T. S. AND BLOSTEIN, S. D. (1987): Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (5), 698–700 73

ASTHANA, A., ZAFEIRIOU, S., CHENG, S. AND PANTIC, M. (2013): Robust Discriminative Response Map Fitting with Constrained Local Models. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, 3444–3451 23

BARNARD, M., HOLDEN, E. J. AND OWENS, R. (2002): Lip tracking using pattern matching snakes. In *Proceedings of the 5th Asian Conference on Computer Vision*, ACCV '02, 1–6 29

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B. AND GROSS, M. (2010): High-quality Single-shot Capture of Facial Geometry. *ACM Trans. Graph.* 29 (4), 40:1–40:9 20, 21

BEELER, T., BICKEL, B., NORIS, G., BEARDSLEY, P., MARSCHNER, S., SUMNER, R. W. AND GROSS, M. (2012): Coupled 3D Reconstruction of Sparse Facial Hair and Skin. *ACM Trans. Graph.* 31 (4), 117:1–117:10 142

BEELER, T. AND BRADLEY, D. (2014): Rigid Stabilization of Facial Expressions. *ACM Trans. Graph.* 33 (4), 44:1–44:9 1, 20, 28

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W. AND GROSS, M. (2011): High-quality Passive Facial Performance Capture Using Anchor Frames. *ACM Trans. Graph.* 30 (4), 75:1–75:10 1, 20, 51, 120, 144, 146, 174

BÉRARD, P., BRADLEY, D., GROSS, M. AND BEELER, T. (2016): Lightweight Eye Capture Using a Parametric Model. *ACM Trans. Graph.* 35 (4), 117:1–117:12 168, 169

BÉRARD, P., BRADLEY, D., NITTI, M., BEELER, T. AND GROSS, M. (2014): High-quality Capture of Eyes. *ACM Trans. Graph.* 33 (6), 223:1–223:12 142, 168

BERGER, M. A., HOFER, G. AND SHIMODAIRA, H. (2011): Carnival–Combining Speech Technology and Computer Animation. *IEEE Computer Graphics and Applications*, 31 (5), 80–89 36

BERMANO, A., BEELER, T., KOZLOV, Y., BRADLEY, D., BICKEL, B. AND GROSS, M. (2015): Detailed Spatio-temporal Reconstruction of Eyelids. *ACM Trans. Graph.* 34 (4), 44:1–44:11 142

BERMANO, A. H., BRADLEY, D., BEELER, T., ZUND, F., NOWROUZEZAHRAI, D., BARAN, I., SORKINE-HORNUNG, O., PFISTER, H., SUMNER, R. W., BICKEL, B. AND GROSS, M. (2014): Facial Performance Enhancement Using Dynamic Shape Space Analysis. *ACM Trans. Graph.* 33 (2), 13:1–13:12 33, 128, 138, 174

BHAT, K. S., GOLDENTHAL, R., YE, Y., MALLET, R. AND KOPERWAS, M. (2013): High Fidelity Facial Animation Capture and Retargeting with Contours. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, 7–14 1, 31

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H. AND GROSS, M. (2007): Multi-scale Capture of Facial Geometry and Motion. *ACM Trans. Graph.* 26 (3), 33:1–33:10 1, 18

BICKEL, B., LANG, M., BOTSCH, M., OTADUY, M. A. AND GROSS, M. (2008): Pose-space Animation and Transfer of Facial Details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, 57–66 33

BISHOP, C. M. (2006): Pattern Recognition and Machine Learning (Information Science and Statistics). 150

BITOUK, D., KUMAR, N., DHILLON, S., BELHUMEUR, P. AND NAYAR, S. K. (2008): Face Swapping: Automatically Replacing Faces in Photographs. *ACM Trans. Graph.* 27 (3), 39:1–39:8 40

BLACK, M. J. AND YACOOB, Y. (1995): Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of the 5th International Conference on Computer Vision*, ICCV '95, 374–381 25

BLANZ, V., BASSO, C., POGGIO, T. AND VETTER, T. (2003): Reanimating faces in images and video. *Comput. Graph. Forum*, 22, 641–650 26, 32, 35, 37

BLANZ, V., SCHERBAUM, K., VETTER, T. AND SEIDEL, H. (2004): Exchanging Faces in Images. *Comput. Graph. Forum*, 23 (3), 669–676 40, 56

BLANZ, V. AND VETTER, T. (1999): A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, 187–194 21, 23, 26, 32, 40, 110, 113

BOERSMA, P. AND WEENINK, D. (2001): PRAAT, a system for doing phonetics by computer. *Glot International*, 5 (9/10), 341–345 95

BORSHUKOV, G. AND LEWIS, J. P. (2003): Realistic Human Face Rendering for "The Matrix Reloaded". In *ACM SIGGRAPH 2003 Sketches & Applications*, SIGGRAPH '03, 16:1–16:1 19, 40, 166

BOUAZIZ, S., WANG, Y. AND PAULY, M. (2013): Online Modeling for Realtime Facial Animation. *ACM Trans. Graph.* 32 (4), 40:1–40:10 2, 9, 10, 21, 22, 27, 32, 37, 110, 111, 114

BRADLEY, D., HEIDRICH, W., POPA, T. AND SHEFFER, A. (2010): High Resolution Passive Facial Performance Capture. *ACM Trans. Graph.* 29 (4), 41:1–41:10 1, 19, 20, 31

BRADSKI, G. AND KAEHLER, A. (2013): Learning OpenCV: Computer Vision in C++ with the OpenCV Library. 2nd edition., ISBN 1449314651, 9781449314651 13

BRAND, M. (1999): Voice Puppetry. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, 21–28 35, 104

BRAND, M. AND BHOTIKA, R. (2001): Flexible flow for 3D nonrigid tracking and shape recovery. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '01, 315–322 26

BREGLER, C., HERTZMANN, A. AND BIERMANN, H. (2000): Recovering non-rigid 3D shape from image streams. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition* Volume 2,, 690–696 23, 24

BREGLER, C., COVELL, M. AND SLANEY, M. (1997): Video Rewrite: Driving Visual Speech with Audio. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, 353–360 41, 42, 101, 102, 168

BREGLER, C. AND KONIG, Y. (1994): "Eigenlips" for robust speech recognition. In *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '94, 669–672 29

BURKE, E. K., GENDREAU, M., HYDE, M. R., KENDALL, G., OCHOA, G., ÖZCAN, E. AND QU, R. (2013): Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64 (12), 1695–1724 93

BURKE, E. K., HYDE, M., KENDALL, G., OCHOA, G., ÖZCAN, E. AND WOODWARD, J. R.; GENDREAU, M. AND POTVIN, J.-Y., EDITORS (2010): A Classification of Hyper-heuristic Approaches., 449–468 93

Cao, C., Bradley, D., Zhou, K. and Beeler, T. (2015): Real-time High-fidelity Facial Performance Capture. *ACM Trans. Graph.* 34 (4), 46:1–46:9 28, 33, 121, 122, 128, 129

Cao, C., Hou, Q. and Zhou, K. (2014a): Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM Trans. Graph.* 33 (4), 43:1–43:10 2, 26, 28, 32, 37, 120, 121, 128

Cao, C., Weng, Y., Lin, S. and Zhou, K. (2013): 3D Shape Regression for Real-time Facial Animation. *ACM Trans. Graph.* 32 (4), 41:1–41:10 26, 32, 37

Cao, C., Weng, Y., Zhou, S., Tong, Y. and Zhou, K. (2014b): FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics*, 20 (3), 413–425 32, 37, 110

Cao, C., Wu, H., Weng, Y., Shao, T. and Zhou, K. (2016): Real-time Facial Animation with Image-based Dynamic Avatars. *ACM Trans. Graph.* 35 (4), 126:1–126:12 168, 169

Cao, Y., Faloutsos, P., Kohler, E. and Pighin, F. (2004): Real-time Speech Motion Synthesis from Recorded Motions. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '04, 345–353 36

Chai, J.-x., Xiao, J. and Hodgins, J. (2003): Vision-based Control of 3D Facial Animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, 193–206 39

Chai, M., Luo, L., Sunkavalli, K., Carr, N., Hadap, S. and Zhou, K. (2015): High-quality Hair Modeling from a Single Portrait Photo. *ACM Trans. Graph.* 34 (6), 204:1–204:10 168

Chai, M., Shao, T., Wu, H., Weng, Y. and Zhou, K. (2016): AutoHair: Fully Automatic Hair Modeling from a Single Image. *ACM Trans. Graph.* 35 (4), 116:1–116:12 168

Chai, M., Zheng, C. and Zhou, K. (2014): A Reduced Model for Interactive Hairs. *ACM Trans. Graph.* 33 (4), 124:1–124:11 168

Chang, Y.-J. and Ezzat, T. (2005): Transferable Videorealistic Speech Animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, 143–151 42

Chong, H. Y., Gortler, S. J. and Zickler, T. (2008): A Perception-based Color Space for Illumination-invariant Image Processing. *ACM Trans. Graph.* 27 (3), 61:1–61:7 58

Chuang, E. and Bregler, C. (2002): Performance-driven Facial Animation using Blend Shape Interpolation. Stanford University (CS-TR-2002-02). – Technical report 37

Cootes, T. F., Taylor, C. J., Cooper, D. H. and Graham, J. (1995): Active Shape Models&Mdash;Their Training and Application. *Comput. Vis. Image Underst.* 61 (1), 38–59 23

Cootes, T. F., Edwards, G. J. and Taylor, C. J. (2001): Active Appearance Models. *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (6), 681–685 23

Cristinacce, D. and Cootes, T. F. (2006): Feature Detection and Tracking with Constrained Local Models. In *Proceedings of the 2006 British Machine Vision Conference*, BMVC '06, 929–938 48

DALE, K., SUNKAVALLI, K., JOHNSON, M. K., VLASIC, D., MATUSIK, W. AND PFISTER, H. (2011): Video Face Replacement. *ACM Trans. Graph.* 30 (6), 130:1–130:10 41, 42, 46, 56, 64, 68, 101, 102

DANTONE, M., GALL, J., FANELLI, G. AND VAN GOOL, L. (2012): Real-time Facial Feature Detection Using Conditional Regression Forests. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, 2578–2585 23

DAVID, P., DEMENTHON, D., DURAISWAMI, R. AND SAMET, H. (2004): SoftPOSIT: Simultaneous Pose and Correspondence Determination. *Int. J. Comput. Vision*, 59 (3), 259–284 73, 115

DECARLO, D. AND METAXAS, D. (1996): The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proceedings of the 1996 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '96, 231–238 25, 26, 27, 38

DENG, Z. AND NEUMANN, U. (2006): eFASE: Expressive Facial Animation Synthesis and Editing with Phoneme-isomap Controls. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, 251–260 36

DI4D: Dimensional Imaging. http://www.di4d.com/ 20

DOLLÁR, P., TU, Z. AND BELONGIE, S. (2006): Supervised Learning of Edges and Object Boundaries. In *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition* Volume 2,, 1964–1971 30, 143, 146

DOLLÁR, P. AND ZITNICK, C. L. (2015): Fast Edge Detection Using Structured Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (8), 1558–1570 30

ECHEVARRIA, J. I., BRADLEY, D., GUTIERREZ, D. AND BEELER, T. (2014): Capturing and Stylizing Hair for 3D Fabrication. *ACM Trans. Graph.* 33 (4), 125:1–125:11 142

EISERT, P. AND GIROD, B. (1998): Analyzing Facial Expressions for Virtual Conferencing. *IEEE Comput. Graph. Appl.* 18 (5), 70–78 38

ESSA, I., BASU, S., DARRELL, T. AND PENTLAND, A. (1996): Modeling, Tracking and Interactive Animation of Faces and Heads Using Input from Video. In *Proceedings of the Computer Animation*, CA '96, 68–79 25, 26

EVENO, N., CAPLIER, A. AND COULON, P. (2004): Accurate and quasi-automatic lip tracking. *IEEE Trans. Circuits Syst. Video Techn.* 14 (5), 706–715 29

EZZAT, T., GEIGER, G. AND POGGIO, T. (2002): Trainable Videorealistic Speech Animation. *ACM Trans. Graph.* 21 (3), 388–398 41

FASEL, B. AND LUETTIN, J. (2003): Automatic facial expression analysis: a survey. *Pattern Recognition*, 36 (1), 259–275 23

FORSYTH, D. A. AND PONCE, J. (2012): Computer Vision: A Modern Approach. 2nd edition. 12

FURUKAWA, Y. AND PONCE, J. (2009): Dense 3D motion capture for human faces. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '09, 1674–1681 18

FYFFE, G., JONES, A., ALEXANDER, O., ICHIKARI, R. AND DEBEVEC, P. (2014): Driving High-Resolution Facial Scans with Video Performance Capture. *ACM Trans. Graph.* 34 (1), 8:1–8:14 27, 28, 33, 106, 129

GARG, R., ROUSSOS, A. AND AGAPITO, L. (2013a): Dense Variational Reconstruction of Non-rigid Surfaces from Monocular Video. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, 1272–1279 28

GARG, R., ROUSSOS, A. AND AGAPITO, L. (2013b): A Variational Approach to Video Registration with Subspace Constraints. *International Journal of Computer Vision*, 104 (3), 286–314 28

GARRIDO, P. AND CASTRO, C. (2012): A Flexible and Adaptive Hyper-heuristic Approach for (Dynamic) Capacitated Vehicle Routing Problems. *Fundamenta Informaticae*, 119 (1), 29–60 93

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X. AND DEBEVEC, P. (2011): Multiview Face Capture Using Polarized Spherical Gradient Illumination. *ACM Trans. Graph.* 30 (6), 129:1–129:10 21

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H. AND PIGHIN, F. (1998): Making Faces. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 55–66 18

HEWER, A., STEINER, I., BOLKART, T., WUHRER, S. AND RICHMOND, K. (2016): A statistical shape space model of the palate surface trained on 3D MRI scans of the vocal tract. *CoRR* abs/1602.07679 167

HEWER, A., STEINER, I. AND WUHRER, S. (2014): A hybrid approach to 3d tongue modeling from vocal tract MRI using unsupervised image segmentation and mesh deformation. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, INTERSPEECH '14, 418–421 167

HIGHAM, N. J. (1986): Computing the Polar Decomposition with Applications. *SIAM J. Sci. Stat. Comput.* 7 (4), 1160–1174 111

HOERL, A. E. AND KENNARD, R. W. (2000): Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 42 (1), 80–86 114, 131, 150

HSIEH, P., MA, C., YU, J. AND LI, H. (2015): Unconstrained realtime facial performance capture. In *Poceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, 1675–1683 22, 32

HU, L., MA, C., LUO, L. AND LI, H. (2015): Single-view Hair Modeling Using a Hairstyle Database. *ACM Trans. Graph.* 34 (4), 125:1–125:9 142, 168

HUANG, H., CHAI, J., TONG, X. AND WU, H.-T. (2011): Leveraging Motion Capture and 3D Scanning for High-fidelity Facial Performance Acquisition. *ACM Trans. Graph.* 30 (4), 74:1–74:10 18

HUANG, X., ZHANG, S., WANG, Y., METAXAS, D. N. AND SAMARAS, D. (2004): A Hierarchical Framework For High Resolution Facial Expression Tracking. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPR Workshops '04, 22 1, 19

ICHIM, A. E., BOUAZIZ, S. AND PAULY, M. (2015): Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Trans. Graph.* 34 (4), 45:1–45:14 34, 106, 109, 111, 124, 129, 138, 168, 169

JONES, A., CHIANG, J., GHOSH, A., LANG, M., HULLIN, M., BUSCH, J. AND DEBEVEC, P. (2008): Real-time Geometry and Reflectance Capture for Digital Face Replacement. University of Southern California (4s). – Technical report 40

JOSHI, P., TIEN, W. C., DESBRUN, M. AND PIGHIN, F. (2003): Learning Controls for Blend Shape Based Realistic Facial Animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, 187–192 128

KASS, M., WITKIN, A. AND TERZOPOULOS, D. (1988): Snakes: Active contour models. *International Journal of Computer Vision*, 1 (4), 321–331 29

KAUCIC, R. AND BLAKE, A. (1998): Accurate, Real-Time, Unadorned Lip Tracking. In *Proceedings of the 1998 International Conference on Computer Vision*, ICCV '98, 370–375 29, 30

KAWAI, M., IWAO, T., MAEJIMA, A. AND MORISHIMA, S. (2015): Automatic Generation of Photorealistic 3D Inner Mouth Animation only from Frontal Images. *Journal of Information Processing*, 23 (5), 693–703 31, 167

KEMELMACHER-SHLIZERMAN, I., SANKAR, A., SHECHTMAN, E. AND SEITZ, S. M. (2010): Being John Malkovich. In *Proceedings of the 11th European Conference on Computer Vision* Volume 6311,, 341–353 42, 49, 52, 68

KEMELMACHER SHLIZERMAN, I. AND SEITZ, S. M. (2011): Face Reconstruction in the Wild. In *Proceedings of the 2011 IEEE International Conference on Computer Vision*, ICCV '11, 1746–1753 27

KEMELMACHER SHLIZERMAN, I., SHECHTMAN, E., GARG, R. AND SEITZ, S. M. (2011): Exploring Photobios. *ACM Trans. Graph.* 30 (4), 61:1–61:10 43, 49

KILBORN, R. (1993): Speak my language: Current attitudes to television subtitling and dubbing. *Media Culture Society*, 15 (4), 641–660 87

KLEHM, O., ROUSSELLE, F., PAPAS, M., BRADLEY, D., HERY, C., BICKEL, B., JAROSZ, W. AND BEELER, T. (2015): Recent Advances in Facial Appearance Capture. *Comput. Graph. Forum*, 34 (2), 709–733 1, 11, 17

KOMOROWSKI, D., MELAPUDI, V., MORTILLARO, D. AND LEE, G. S. (2010): A Hybrid Approach to Facial Rigging. In *ACM SIGGRAPH ASIA 2010 Sketches*, SA '10, 42:1–42:2 1, 11

KSHIRSAGAR, S. AND MAGNENAT-THALMANN, N. (2003): Visyllable Based Speech Animation. *Comput. Graph. Forum*, 22 (3), 632–640 35

LEGOFF, B., GUIARD-MARIGNY, T., COHEN, M. M. AND BENOÎT, C. (1994): Real-time analysis-synthesis and intelligibility of talking faces. In *Proceedings of the 2nd ESCA/IEEE Workshop on Speech Synthesis*, 53–56 29, 87

LESNER, S. A. AND KRICOS, P. B. (1981): Visual vowel and diphthong perception across speakers. *Journal of the Academy of Rehabilitative Audiology*, 14, 252–258 87

LEVENBERG, K. (1944): A Method for the Solution of Certain Non-Linear Problems in Least Squares. *The Quarterly of Applied Mathematics*, 2 (2), 164–168 115

LÉVY, B. AND ZHANG, H. R. (2010): Spectral Mesh Processing. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, 8:1–8:312 111

LEWIS, J. P., ANJYO, K., RHEE, T., ZHANG, M., PIGHIN, F. AND DENG, Z. (2014): Practice and Theory of Blendshape Facial Models. In LEFEBVRE, S. AND SPAGNUOLO, M., EDITORS: *Eurographics 2014 - State of the Art Reports*, 199–218 9, 10, 32, 93, 128

LI, C., ZHOU, K. AND LIN, S. (2014): Intrinsic Face Image Decomposition with Human Face Priors. In *Proceedings of the 13th European Conference on Computer Vision* Volume 8693,, 218–233 166

LI, C., ZHOU, K. AND LIN, S. (2015a): Simulating makeup through physics-based manipulation of intrinsic image layers. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, 4621–4629 166

LI, G., WU, C., STOLL, C., LIU, Y., VARANASI, K., DAI, Q. AND THEOBALT, C. (2013a): Capturing Relightable Human Performances under General Uncontrolled Illumination. *Comput. Graph. Forum*, 32 (2), 275–284 166

LI, H., ROIVAINEN, P. AND FORCHEIMER, R. (1993): 3DD Motion Estimation in Model-Based Facial Image Coding. *IEEE Trans. Pattern Anal. Mach. Intell.* 15 (6), 545–555 25

LI, H., WEISE, T. AND PAULY, M. (2010): Example-based Facial Rigging. *ACM Trans. Graph.* 29 (4), 32:1–32:6 9, 21, 34, 70, 129

LI, H., YU, J., YE, Y. AND BREGLER, C. (2013b): Realtime Facial Animation with On-the-fly Correctives. *ACM Trans. Graph.* 32 (4), 42:1–42:10 2, 9, 10, 22, 26, 27, 30, 32, 37, 110

LI, J., XU, W., CHENG, Z., XU, K. AND KLEIN, R. (2015b): Lightweight wrinkle synthesis for 3D facial modeling and animation. *Computer-Aided Design*, 58, 117–122 33, 129

LI, K., XU, F., WANG, J., DAI, Q. AND LIU, Y. (2012): A data-driven approach for facial expression synthesis in video. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, 57–64 43, 52, 61, 68

LIU, K. AND OSTERMANN, J. (2011): Realistic facial expression synthesis for an image-based talking head. In *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo*, ICME '11, 1–6 42

LIU, Y., XU, F., CHAI, J., TONG, X., WANG, L. AND HUO, Q. (2015): Video-audio Driven Real-time Facial Animation. *ACM Trans. Graph.* 34 (6), 182:1–182:10 31

LIU, Z., SHAN, Y. AND ZHANG, Z. (2001): Expressive Expression Mapping with Ratio Images. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 271–276 40, 42, 43

LUO, L., LI, H., PARIS, S., WEISE, T., PAULY, M. AND RUSINKIEWICZ, S. (2012): Multi-view hair capture using orientation fields. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, 1490–1497 142

MA, J., COLE, R. A., PELLOM, B. L., WARD, W. H. AND WISE, B. (2006): Accurate Visible Speech Synthesis Based on Concatenating Variable Length Motion Capture Data. *IEEE Trans. Vis. Comput. Graph.* 12 (2), 266–276 35, 36, 168

MA, W.-C., HAWKINS, T., PEERS, P., CHABERT, C.-F., WEISS, M. AND DEBEVEC, P. (2007): Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR '07, 183–194 20, 21, 40

MA, W.-C., JONES, A., CHIANG, J.-Y., HAWKINS, T., FREDERIKSEN, S., PEERS, P., VUKOVIC, M., OUHYOUNG, M. AND DEBEVEC, P. (2008): Facial Performance Synthesis Using Deformation-driven Polynomial Displacement Maps. *ACM Trans. Graph.* 27 (5), 121:1–121:10 33

MARQUARDT, D. W. (1963): An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11 (2), 431–441 115

MCGURK, H. AND MACDONALD, J. (1976): Hearing lips and seeing voices. *Nature*, 264 (5588), 746–748 87

MORÉ, J. J.; WATSON, G. A., EDITOR (1978): Lecture Notes in Mathematics. Volume 630, The Levenberg-Marquardt algorithm: Implementation and theory., 105–116 115

MOVA: MOVA®Contour®Facial Capture System. http://rearden.com/mova.html 18

MÜLLER, C. (1966): Spherical harmonics. Volume 17, Lecture Notes in Mathematics. 109

NATH, A. R. AND BEAUCHAMP, M. S. (2012): A neural basis for interindividual differences in the McGurk effect, a multisensory speech illusion. *NeuroImage*, 59 (1), 781–787 142

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J. AND RAMAMOORTHI, R. (2005): Efficiently Combining Positions and Normals for Precise 3D Geometry. *ACM Trans. Graph.* 24 (3), 536–543 78, 115

NEUMANN, T., VARANASI, K., WENGER, S., WACKER, M., MAGNOR, M. AND THEOBALT, C. (2013): Sparse Localized Deformation Components. *ACM Trans. Graph.* 32 (6), 179:1–179:10 128

NGUYEN, Q. D. AND MILGRAM, M. (2009): Semi Adaptive Appearance Models for lip tracking. In *Proceedings of the International Conference on Image Processing*, ICIP, 2437–2440 30

NOH, J.-Y. AND NEUMANN, U. (2001): Expression Cloning. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 277–288 38, 39

OJALA, T., PIETIKÄINEN, M. AND MÄENPÄÄ, T. (2002): Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (7), 971–987 49

OWENS, E. AND BLAZEK, B. (1986): Visemes observed by the hearing-impaired and normal-hearing adult viewers. *Journal of Speech and Hearing Research*, 28, 381–393 86, 87

PALADINI, M., BARTOLI, A. AND AGAPITO, L. (2010): Sequential Non-rigid Structure-from-motion with the 3D-implicit Low-rank Shape Model. In *Proceedings of the 11th European Conference on Computer Vision* Volume 6312,, 15–28 24

PALADINI, M., DEL BUE, A., STOSIC, M., DODIG, M., XAVIER, J. M. F. AND AGAPITO, L. (2009): Factorization for non-rigid and articulated structure using metric projections. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '09, 2898–2905 24

PAPENBERG, N., BRUHN, A., BROX, T., DIDAS, S. AND WEICKERT, J. (2006): Highly accurate optic flow computation with theoretically justified warping. *Int. J. of Comput. Vision*, 67 (2), 141–158 18, 77

PÉREZ, P., GANGNET, M. AND BLAKE, A. (2003): Poisson Image Editing. *ACM Trans. Graph.* 22 (3), 313–318 57, 58

PIGHIN, F. AND LEWIS, J. (2006): Performance-Driven Facial Animation. In *ACM SIGGRAPH Courses* 17

PIGHIN, F., SZELISKI, R. AND SALESIN, D. (1999): Resynthesizing facial animation through 3D model-based tracking. In *Proceedings of the 7th International Conference on Computer Vision*, ICCV '99, 143–150 25

PLATT, J. C. (1998): Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research (MSR-TR-98-14). – Technical report 73

RAMAMOORTHI, R. AND HANRAHAN, P. (2001): A Signal-processing Framework for Inverse Rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 117–128 109

SAITO, S., LI, T. AND LI, H. (2016): Real-Time Facial Segmentation and Performance Capture from RGB Input. In *Proceedings of the 14th European Conference on Computer Vision* Volume 9912,, 244–261 26, 32, 167

SARAGIH, J. M., LUCEY, S. AND COHN, J. F. (2011a): Deformable Model Fitting by Regularized Landmark Mean-Shift. *Int. J. Comput. Vision*, 91 (2), 200–215 23, 24, 48, 56, 65, 68, 72, 113, 147

SARAGIH, J. M., LUCEY, S. AND COHN, J. F. (2011b): Real-time avatar animation from a single image. In *Proceedings of the Ninth IEEE International Conference on Automatic Face and Gesture Recognition*, FG '11, 117–124 37

SEOL, Y., LEWIS, J., SEO, J., CHOI, B., ANJYO, K. AND NOH, J. (2012): Spacetime Expression Cloning for Blendshapes. *ACM Trans. Graph.* 31 (2), 14:1–14:12 103

SHI, F., WU, H.-T., TONG, X. AND CHAI, J. (2014): Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM Trans. Graph.* 33 (6), 222:1–222:13 27, 109, 117, 122, 123, 124, 128, 174

SIFAKIS, E., SELLE, A., ROBINSON-MOSHER, A. AND FEDKIW, R. (2006): Simulating Speech with a Physics-based Facial Muscle Model. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, 261–270 32, 35

SLANEY, M. AND COVELL, M. (2000): FaceSync: A Linear Operator for Measuring Synchronization of Video Facial Images and Audio Tracks. In *Proceedings of Neural Information Processing Systems*, NIPS '00, 814–820 87

SONG, M., DONG, Z., THEOBALT, C., WANG, H., LIU, Z. AND SEIDEL, H. P. (2007): A Generic Framework for Efficient 2-D and 3-D Facial Expression Analogy. *IEEE Trans. Multimedia*, 9 (7), 1384–1395 38

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C. AND SEIDEL, H.-P. (2004): Laplacian Surface Editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, 175–184 18, 38

SORKINE, O. (2005): Laplacian Mesh Processing. In CHRYSANTHOU, Y. AND MAGNOR, M. A., EDITORS: *Eurographics 2005 - State of the Art Reports*, 53–70 70, 77

SUMBY, W. H. AND POLLACK, I. (1954): Visual Contribution to Speech Intelligibility in Noise. *The Journal of the Acoustical Society of America*, 26 (2), 212–215 86, 87

SUMMERFIELD, Q. (1992): Lipreading and Audio-Visual Speech Perception. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 335 (1273), 71–78 29, 86, 87

SUMNER, R. W. AND POPOVIĆ, J. (2004): Deformation Transfer for Triangle Meshes. *ACM Trans. Graph.* 23 (3), 399–405 19, 22, 34, 38, 110, 111, 133, 147, 157

SUWAJANAKORN, S., SEITZ, S. M. AND KEMELMACHER-SHLIZERMAN, I. (2015): What Makes Tom Hanks Look Like Tom Hanks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, ICCV '15, 3952–3960 39

SUWAJANAKORN, S., SHLIZERMAN, I. K. AND SEITZ, S. M. (2014): Total Moving Face Reconstruction. In *Proceedings of the 13th European Conference on Computer Vision* Volume 8692,, 796–812 27, 109, 124, 135

TAN, X. AND TRIGGS, B. (2010): Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Trans. Image Processing*, 19 (6), 1635–1650 49

TAYLOR, S. L., MAHLER, M., THEOBALD, B.-J. AND MATTHEWS, I. (2012): Dynamic Units of Visual Speech. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, 275–284 35, 87, 168

TENA, J. R., TORRE, F. DE LA AND MATTHEWS, I. (2011): Interactive Region-based Linear 3D Face Models. *ACM Trans. Graph.* 30 (4), 76:1–76:10 32

THEOBALD, B.-J., MATTHEWS, I., MANGINI, M., SPIES, J. R., BRICK, T. R., COHN, J. F. AND BOKER, S. M. (2009): Mapping and manipulating facial expression. *Language and Speech*, 52 (2–3), 369–386 37, 91, 103

THIES, J., ZOLLHÖFER, M., STAMMINGER, M., THEOBALT, C. AND NIESSNER, M. (2016): Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16 2, 32, 43, 101, 164, 165

THIES, J., ZOLLHÖFER, M., NIESSNER, M., VALGAERTS, L., STAMMINGER, M. AND THEOBALT, C. (2015): Real-time Expression Transfer for Facial Reenactment. *ACM Trans. Graph.* 34 (6), 183:1–183:14 2, 9, 10, 14, 15, 22, 32, 43, 118

TIAN, Y.-L., KANADE, T. AND COHN, J. F. (2000): Robust Lip Tracking by Combining Shape, Color and Motion. In *Proceedings of the 3rd Asian Conference on Computer Vision*, ACCV '00, 1–6 29

VALGAERTS, L., BRUHN, A., MAINBERGER, M. AND WEICKERT, J. (2012a): Dense Versus Sparse Approaches for Estimating the Fundamental Matrix. *Int. J. Comput. Vision*, 96 (2), 212–234 70

VALGAERTS, L., BRUHN, A., ZIMMER, H., WEICKERT, J., STOLL, C. AND THEOBALT, C. (2010): Joint Estimation of Motion, Structure and Geometry from Stereo Sequences. In *Proceedings of the 11th European Conference on Computer Vision* Volume 6314,, 568–581 76

VALGAERTS, L., WU, C., BRUHN, A., SEIDEL, H.-P. AND THEOBALT, C. (2012b): Lightweight Binocular Facial Performance Capture Under Uncontrolled Lighting. *ACM Trans. Graph.* 31 (6), 187:1–187:11 14, 20, 21, 51, 68, 69, 76, 77, 78, 82, 83, 84, 106, 115, 118, 120, 124, 162, 173

VALLET, B. AND LÉVY, B. (2008): Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum*, 27 (2), 251–260 111

VICON: Vicon Motion Systems Ltd. https://www.vicon.com 18

VIOLA, P. AND JONES, M. J. (2004): Robust Real-Time Face Detection. *Int. J. Comput. Vision*, 57 (2), 137–154 24, 48

VLASIC, D., BRAND, M., PFISTER, H. AND POPOVIĆ, J. (2005): Face Transfer with Multilinear Models. *ACM Trans. Graph.* 24 (3), 426–433 26, 32, 37, 41, 68

VOLZ, S., BRUHN, A., VALGAERTS, L. AND ZIMMER, H. (2011): Modeling temporal coherence for optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV '11, 1116–1123 77

WANG, C., SHI, F., XIA, S. AND CHAI, J. (2016): Realtime 3D Eye Gaze Animation Using a Single RGB Camera. *ACM Trans. Graph.* 35 (4), 118:1–118:14 168

WANG, Y., HUANG, X., LEE, C.-S., ZHANG, S., LI, Z., SAMARAS, D., METAXAS, D., ELGAMMAL, A. AND HUANG, P. (2004): High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. *Computer Graphics Forum*, 23 (3), 677–686 1, 19, 30

WEISE, T., BOUAZIZ, S., LI, H. AND PAULY, M. (2011): Realtime Performance-based Facial Animation. *ACM Trans. Graph.* 30 (4), 77:1–77:10 9, 21, 22, 32, 37, 68, 106, 129

WEISE, T., LEIBE, B. AND GOOL, L. J. V. (2007): Fast 3D scanning with automatic motion compensation. In *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '07, 1–8 19, 37

WEISE, T., LI, H., VAN GOOL, L. AND PAULY, M. (2009): Face/Off: Live Facial Puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, 7–16 1, 19, 37, 56

WILLIAMS, L. (1990): Performance-driven Facial Animation. *SIGGRAPH Comput. Graph.* 24 (4), 235–242 18

WILSON, C. A., GHOSH, A., PEERS, P., CHIANG, J.-Y., BUSCH, J. AND DEBEVEC, P. (2010): Temporal Upsampling of Performance Geometry Using Photometric Alignment. *ACM Trans. Graph.* 29 (2), 17:1–17:11 20

WU, C., BRADLEY, D., GROSS, M. AND BEELER, T. (2016): An Anatomically-constrained Local Deformation Model for Monocular Face Capture. *ACM Trans. Graph.* 35 (4), 115:1–115:12 28, 135

WU, C., STOLL, C., VALGAERTS, L. AND THEOBALT, C. (2013): On-set Performance Capture of Multiple Actors with a Stereo Camera. *ACM Trans. Graph.* 32 (6), 161:1–161:11 15

WU, C., VARANASI, K., LIU, Y., SEIDEL, H. AND THEOBALT, C. (2011a): Shading-based dynamic shape refinement from multi-view video under general illumination. In *Proceedings of the 2011 IEEE International Conference on Computer Vision*, ICCV '11, 1108–1115 20

WU, C., WILBURN, B., MATSUSHITA, Y. AND THEOBALT, C. (2011b): High-quality Shape from Multi-view Stereo and Shading Under General Illumination. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, 969–976 14

WU, C., ZOLLHÖFER, M., NIESSNER, M., STAMMINGER, M., IZADI, S. AND THEOBALT, C. (2014): Real-time Shading-based Refinement for Consumer Depth Cameras. *ACM Trans. Graph.* 33 (6), 200:1–200:10 118, 164

XIAO, J., BAKER, S., MATTHEWS, I. AND KANADE, T. (2004): Real-time Combined 2D+3D Active Appearance Models. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'04, 535–542 23

XIONG, X. AND TORRE, F. D. L. (2013): Supervised Descent Method and Its Applications to Face Alignment. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, 532–539 24

YEHIA, H., RUBIN, P. AND VATIKIOTIS-BATESON, E. (1998): Quantitative association of vocal-tract and facial behavior. *Speech Communication*, 26 (1-2), 23–43 36, 87

YOUNG, S., EVERMANN, G., GALES, M., HAIN, T., KERSHAW, D., LIU, X. A., MOORE, G., ODELL, J., OLLASON, D., POVEY, D., VALTCHEV, V. AND WOODLAND, P. (2006): The HTK Book. 95

YPSILOS, I. A., HILTON, A., TURKMANI, A. AND JACKSON, P. J. B. (2004): Speech-Driven Face Synthesis from 3D Video. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission*, 3DPVT '04, 58–65 167

ZAFEIRIOU, S., ZHANG, C. AND ZHANG, Z. (2015): A Survey on Face Detection in the Wild. *Comput. Vis. Image Underst.* 138 (C), 1–24 23

ZHANG, L., SNAVELY, N., CURLESS, B. AND SEITZ, S. M. (2004): Spacetime Faces: High Resolution Capture for Modeling and Animation. *ACM Trans. Graph.* 23 (3), 548–558 18

ZHANG, Z. (2000): A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (11), 1330–1334 13

ZHU, X. AND RAMANAN, D. (2012): Face Detection, Pose Estimation, and Landmark Localization in the Wild. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, 2879–2886 24

ZOLLHÖFER, M., NIESSNER, M., IZADI, S., REHMANN, C., ZACH, C., FISHER, M., WU, C., FITZGIBBON, A., LOOP, C., THEOBALT, C. AND STAMMINGER, M. (2014): Real-time Non-rigid Reconstruction Using an RGB-D Camera. *ACM Trans. Graph.* 33 (4), 156:1–156:12 113, 116, 118, 164