Diss. ETH No. 23032

Geometric Methods for Realistic Animation of Faces

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by
Amit H. Bermano
MSc in Computer Science, The Technion - Israel Institue of Technology, Israel
born 20.11.1980
citizen of Israel

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Prof. Dr. Olga Sorkine-Hornung, co-examiner
Prof. Dr. Bernd Bickel, co-examiner

2015

Abstract

Realistic facial synthesis is one of the most fundamental problems in computer graphics, and has been sought after for approximately four decades. It is desired in a wide variety of fields, such as character animation for films and advertising, computer games, video teleconferencing, user-interface agents and avatars, and facial surgery planning. Humans, on the other hand, are experts in identifying every detail and every regularity or variation in proportion from one individual to the next. The task of creating a realistic human face is elusive due to this, as well as many other factors. Among which are complex surface details, spatially and temporally varying skin texture and subtle emotions that are conveyed through even more subtle motions.

In this thesis, we present the most commonly practiced facial content creation process, and contribute to the quality of each of its steps. The proposed algorithms significantly increase the level of realism attained by each step and therefore substantially reduce the amount of manual labor required for production quality facial content. The thesis contains three parts, each contributing to one step of the facial content creation pipeline.

In the first part, we aim at greatly increasing the fidelity of facial performance captures, and present the first method for detailed spatio-temporal reconstruction of eyelids. Easily integrable with existing high quality facial performance capture approaches, this method generates a person-specific, time-varying eyelid reconstruction with anatomically plausible deformations. Our approach is to combine a geometric deformation model with image data, leveraging multi-view stereo, optical flow, contour tracking and wrinkle detection from local skin appearance. Our deformation model serves as a prior that enables reconstruction of eyelids even under strong self-occlusions caused by rolling and folding skin as the eye opens and closes.

In the second part, we contribute to the authoring step of the creation process. We present a method for adding fine-scale details and expressiveness to low-resolution art-directed facial performances. Employing a high-resolution facial performance capture system, we augment artist friendly content, such as those created manually using a rig, via marker-based capture, by fitting a morphable model to a video, or through Kinect-based reconstruction. From the high fidelity

captured data, our system encodes subtle spatial and temporal deformation details specific to that particular individual, and composes the relevant ones to the desired input animation. The resulting animations exhibit compelling animations with nuances and fine spatial details that match captured performances, while preserving the artistic intent authored by the low-resolution input sequences, outperforming current state-of-the-art in example-based facial animation.

The third part of the dissertation proposes to enrich digital facial content by adding a significant sense of presence. Replacing the classic 2D or 3D displaying techniques of digital content, we propose the first complete process for augmenting deforming physical avatars using projector-based illumination. Physical avatars have been long used to give physical presence to a character, both in the field of entertainment and teleconferencing. Using a human-shaped display surface provides depth cues and multiple observers with their own perspectives. Such physical avatars, however, suffer from limited movement and expressiveness due to mechanical constraints. Given an input animation, our system decomposes the motion into low-frequency motion that can be physically reproduced by a robotic head and high-frequency details that are added using projected shading. The result of our system is a highly expressive physical avatar that features facial details and motion otherwise unattainable due to physical constraints.

Zusammenfassung

Realistische Gesichtssynthese ist eines der fundamentalen Probleme in der Computer Graphik, woran seit Jahrzehnten gearbeitet wird. Gesichtssynthese wird in verschiedenen Gebieten angewendet, zum Beispiel in der Charakteranimation für Filme und Werbung, Computer Spiele, Video Telekonferenzen, User-Interface Agenten und Avatare und Gesichtsoperationsplanung. Menschen sind Experten darin, jedes Detail und jede Irregularität in den Proportionen des Gesichts zu erkennen. Somit ist die Aufgabe, ein realistische Gesicht zu kreieren, sehr schwierig. Dazu gehören komplexe Oberflächendetails, räumlich und zeitlich sich verändernde Hauttexturen und subtile Emotionen, die durch sogar noch subtilere Bewegungen übermittelt werden. In dieser Arbeit präsentieren wir den gebräuchlichsten Prozess für das Kreieren von Gesichtern und leisten einen Beitrag zur Qualität für jeden Schritt. Die vorgeschlagenen Algorithmen verbessern den Realismus in jedem Schritt erheblich und verringern somit die Handarbeit, die nötig ist, um Produktionsqualität zu erreichen. Die Arbeit ist in drei Teile aufgeteilt, wobei jeder Teil zu einem Schritt im Gesichtskreationsprozess beiträgt.

Der erste Teil zielt darauf, die Genauigkeit von Gesichtsdarbietungen erheblich zu verbessern und präsentiert die erste Methode für eine detaillierte räumlich-zeitliche Rekonstruktion von Augenlidern. Die Methode ist einfach integrierbar in existierende Ansätzen für hochqualitative Gesichtsdarbietungen. Sie generiert eine personen-spezifische und zeitabhängige Rekonstruktion von Augenlidern mit anatomisch plausiblen Deformationen. Unser Vorgehen besteht darin, ein geometrisches Deformationsmodell mit Bilderdaten zu kombinieren und dabei die Stereoperspektive, optischer Fluss, Kantenverfolgung und Faltenerkennung auszunutzen. Unser Deformationsmodell ermöglicht uns die Rekonstruktion von Augenlidern auch mit starken Selbstokklusionen, die durch rollende und faltende Haut während das Auge sich öffnet und schliesst, entstehen.

Im zweiten Teil tragen wir zum verfassenden Schritt im Gesichtskreationsprozess bei. Wir präsentieren eine Methode um feine Details und Ausdrucksfähigkeit zu handgemachten Gesichtsdarbietungen mit nur tiefer Auflösung hinzuzufügen. Wir benutzen ein hochauflösendes Gesichtsdarbietungsaufnahmesystem um künstlerische Inhalte zu ergänzen. Das System unterstützt Inhalte, die von Hand gemacht wurden mittels eines Rigs, markerbasierende Aufnahmen, durch einpassen eines wandelbaren Modells von einem Video erstellte Inhalte und Kinect-

basierte Rekonstruktionen. Von den hochauflösenden Aufnahmedaten enkodiert unser System subtile räumliche und zeitliche Deformationsdetails für das spezifische Individuum. Die resultierenden Animationen zeigen überzeugende Animationen mit Nuancen und feinen Details, die den Bewegungsaufnahmen entsprechen. Dabei bleibt die künstlerische Absicht der tiefauflösenden Eingangssequenzen erhalten. Sie übertreffen die zur Zeit modernsten beispielbasierten Gesichtsanimationen.

Der dritte Teil der Dissertation befasst sich mit dem Bereichern von digitalen Gesichtern durch hinzufügen eines signifikanten Präsenzgefühlts. Wir beabsichtigen 2D oder 3D Bildschirmtechnologien für digitalen Inhalt mit einem Prozess zur Augmentation von deformierbaren physischen Avataren mit projektorbasierter Illumination zu ersetzen. Physische Avatare werden schon lange verwendet, um einem Wesen physische Präsenz zu verleihen, in der Unterhaltung sowie für Telekonferenzen. Eine menschlich geformte Anzeige birgt Tiefenanhaltspunkte und bietet mehreren Beobachtern eine eigene Perspektive. Jedoch leiden solche physische Avatare an eingeschränkter Bewegungsfreiheit und Ausdrucksfähigkeit aufgrund mechanischer Beschränkungen. Unser System zerlegt die Eingangsanimation in tieffrequente Bewegungen, welche ein Roboterkopf ausführen kann, und hochfrequente Details, welche mit Projektion hinzugefügt werden. Das Resultat ist ein hoch ausdrucksfähiger physischer Avatar, der Gesichtsdetails sowie Bewegungen umfasst, die unerreichbar wären mit einem rein mechanischen Avatar.

Acknowledgments

My thanks go firstly to my advisor Prof. Markus Gross, for creating a wonderful workspace which promotes creative thinking and exchanging ideas. The lab he created with his passion for research is the infrastructure of success for so many students, and provides the atmosphere and means to push the boundaries of human knowledge in the fields of visual computing.

I would also like to thank my co-examiners, Prof. Dr. Olga-Sorkine and Prof. Dr. Bernd Bickel, who have served as examiners, guides and friends throughout my Doctoral studies. It was a pleasure working with you. Thank you both for your patience, ideas, hours of discussions and friendship.

I wish to thank all my collaborators, Prof. Hanspeter pfister, Prof. Dr. Bob Sumner, Prof. Daisuke Iwai, Prof. Derek Nowrouzezahrai, Dr. Thabo Beeler, Dr. Derek Bradley, Dr. Ilya Baran, Fabio Zünd, Yeara Kozlov, Pascal Bérard and Philipp Brüschweiler, for theirs support and guidance, and for helping bringing all the projects to success.

Also, I would like to thank Fabio Zünd for translating the abstract, Alex Chapiro for helping the thesis title, and all the performers and people that helped with generating the results and comparisons: Maya Sigron, Kaan Yücer, and Pascal Bérard, Sean Sutton, Kristen Vermilyea, Hao Li, Volker Heizle, Kevin Dale, Alex Ma, Paul Debevec. A special and warm thanks go to Maurizio Nitti and Alessia Marra for their amazingly skillful touch, bringing the Disney magic to anything they design.

Last but not least, I would like to thank my family and friends for baring with me through the difficult and challenging process. Especially to my wife, Yael Bermano, who believed in me more than I did, encouraged, supported and loved. This would not have happened without you. I am eternally grateful for your warmth and understanding.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

As humans, we are very sensitive to subtle properties of facial features and actions [Smi+05; PE12]. We locate, identify and distinguish between faces with very casual inspection. For example, this is one of the reasons early postage stamps had human faces (of Queen Victoria) on them. It was assumed that it would be easier to detect forgeries. Facial expressions provide information about emotions (such as fear, anger, enjoyment, surprise, sadness, disgust), cognitive activity (such as perplexity, concentration, or boredom), temperament, personality and much more [Ekm+93]. A careful examination of facial expressions over time can also reveal leakage of concealed emotions or truthfulness. In education, the teacher's facial expressions influence whether the pupils learn and the pupil's facial expressions can inform the teacher of the need to adjust the instructional message, even in virtual environments [SJ13]. In business, facial expressions are important in negotiations and personnel decisions. In medicine, facial expressions can be useful in studies of the autonomic nervous system and the psychological state of the patient. In man-machine interactions, facial expressions could provide a way to communicate basic information about needs and demands to computers or encourage specific reactions from an observer.

For these reasons and others, displaying and augmenting faces in a photo realistic manner with computer graphics has been a central goal of the field for over forty years [Par74]. A variety of notable efforts has been made to

create realistic digital actors over the last two decades, each leveraging numerous advances in computer graphics technology and artistry, both in the film industry (*Final Fantasy* (2001), *Spider Man 2* (2004), *The Polar Express* (2004), *Beowulf* (2007), *The Curious Case of Benjamin Button* (2008), *A Christmas Carol* (2009), etc.) and by the academic graphics community [CP99; Sif+06; Ale+10; Ale+13]. Methods that were used in these efforts include traditional photography, highly accurate laser scanners and elaborate and ever evolving light stages to capture the face geometry. Anatomically inspired physically based simulation, hand animated rigs and motion capture were used to drive the motion.

Unfortunately, as accurate and human like as these virtual figures may be, it is well known that when such characters approach realistic similarity to humans, they stop being likable and instead become eerie, frightening, repulsive—"uncanny". Reports indicate that realistic animated characters, like those in the movie *The Polar Express* (2004) had negative effects on viewers, who addressed to the uncomfortable realism of the characters [Gel08]. This phenomena was observed also with regard to robotic figures, such as *Gemenoids* [NIH07b]. Indeed, Mori et al.[1970] hypothesized that as a machine acquires greater similarity to a human, it becomes more emotionally appealing to the observer. However, when it becomes too similar to a human there is a very strong drop in believability and comfort, before finally achieving full humanity and eliciting positive reactions once more. This drop is what he coined as "the Uncanny Valley", and he introduced a hypothetical graph describing the relation between the level of realism and feeling of pleasantness, or familiarity (see Figure1.1).

This phenomena is not yet fully understood, and is considered potentially multi-dimensional [MI06]. The reasons of which this phenomena stems from are also uncertain. Some suggest, based on theories by Sigmond Freud, that human-like robots may be unnerving because they remind people of death [MI06]. Indeed, horror tales throughout history have featured characters such as vampires, zombies and devils - all of which almost human but lack the spark of life. Others suggest that this unnervingness is an evolutionary defense mechanism, designed to detect and avoid the ill. Facial features, such as bizarre eyes, are also considered to be a cause [SN07]. Finally, some claim that the cause is a perceptual mismatch - the appearance and motion of a near-human agent do not match. Saygin et al. [2011] have conducted a research where they had shown subjects three sets of videos: one of human-like robots (or *androids*) performing everyday tasks, one of the same actions as performed by the human on whom the android was modeled and one of a stripped version of the android - without skin and eyes so that it could no longer be mistaken for a human. Measuring brain activity during this

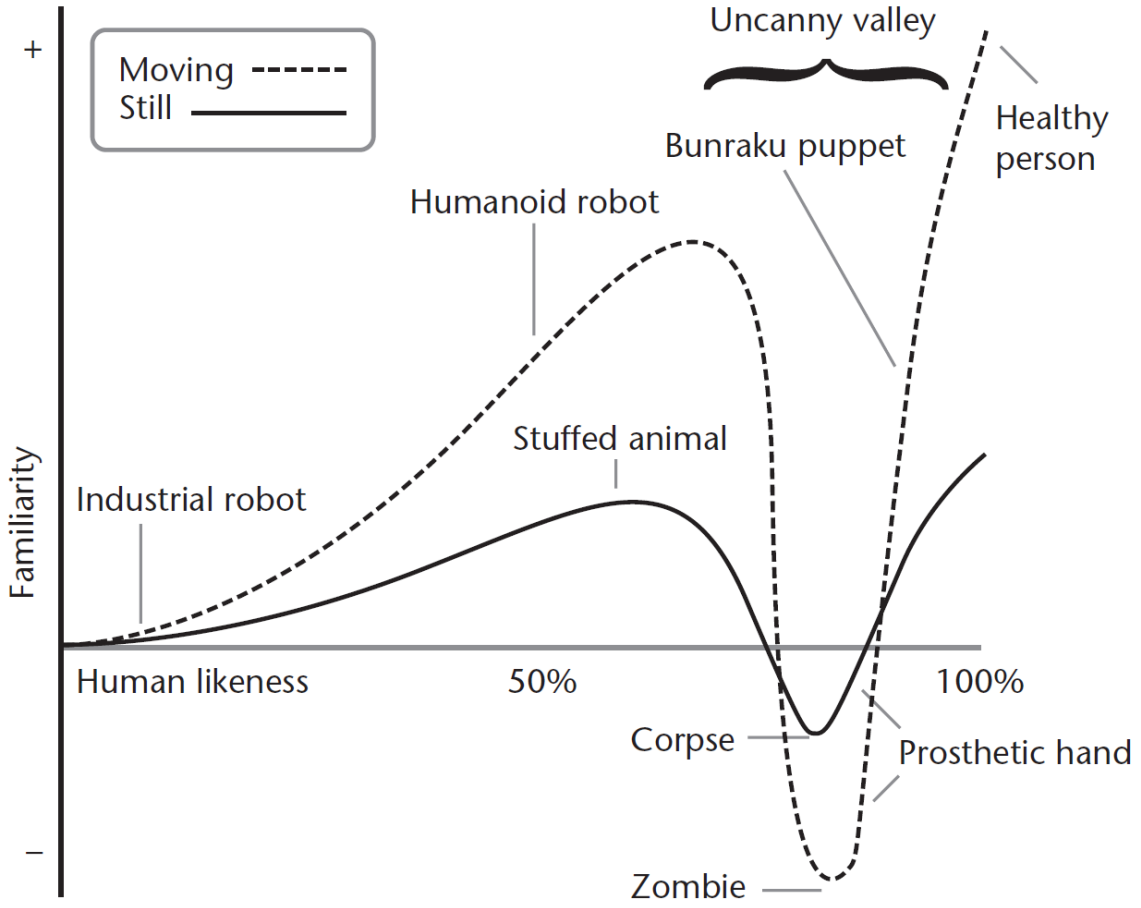


Figure 1.1: *Hypothesized response of humans to human likeness of characters. The uncanny valley is the region of negative response to characters that seem highly human-like (i.e., zombie and corpse). Movement is claimed to emphasize the response for all characters and in particular within the uncanny valley [MMK12]*

process, the researchers have concluded that while the android used in the study was often mistaken for a human at first sight, longer exposure and dynamic viewing have invoked the eeriness attributed to the uncanny valley. This is in correlation with the well established hypothesis that the brain is finely tuned to recognizing biological motions and distinguishing between very subtle characteristics of this motion. In fact, it is believed that the survival of many species is dependent on their ability to recognize complex movements (e.g., of predators, prey and potential mates) [GP03].

This phenomena stresses the fact that in order to achieve believable facial realism, very high accuracy is required both in the spatial and the temporal domains. This could be achieved mainly in two ways. First, by using

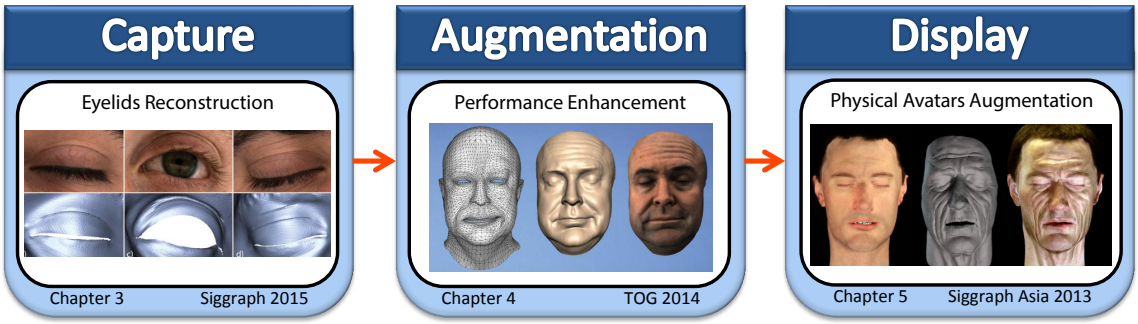


Figure 1.2: *The facial content creation pipeline and our contributions to each of its steps*

a physically and anatomically accurate model. Unfortunately to date, even though the underlying physical models are quite well understood, the human face is an extremely complex bio-mechanical system that is very difficult to model, which largely limits the applicability of such an approach. Traditionally, in order to alleviate such problems, data-driven methods are employed. Data-driven methods offer an alternative to complex models, as they exploit a system's response under several example conditions to approximate its behavior in previously untrained states [Ota+12]. The second approach to achieving the level of realism that crosses the uncanny valley depends on highly accurate facial scanning.

1.2 Facial content creation process

In this thesis, we will explore the process of facial content creation using the latter, more popular, approach. We will elaborate on each step and enhance the realism of it. As depicted in Figure 1.2, the facial content creation process consists of the following steps:

- Capture
- Augmentation
- Display

1.2.1 Capture

The first step to creating digital facial content is to capture both shape and motion of the human face from real life performers. As discussed in Section 1.1, this is the case due to the complexity of the face and the human sensitivity to it, which render manual creation and animation impractical. For shape

capturing, many methods exist varying in technology and traits. The main available techniques can be split into two groups: active or passive captures.

Active capture devices include an emitting part, which is then sensed by its receiving counterpart, while passive ones consist only of a receiver. Most common active capture methods employ laser scanners [BPG04], Time-of-Flight (ToF) sensors [Bü+05], or structured light produced by projectors [HZ04]. All of these methods emit light in a carefully designed way and measure depth by the light that is reflected back to the sensors. These methods are typically very accurate, but suffer from a few inherent drawbacks. First, active methods are less suited for dynamics scenes since they typically require several iterations of emitting and receiving for complete depth measurement. Second, active methods rely on the light that is reflected from the measured object, which impairs accuracy in the case of translucent materials, such as the skin. Third, texture has to be captured in a different way since color is not captured by any of these devices.

Passive methods, also called passive stereo, employ only cameras. Depth is computed by triangulating corresponding points in the captured images, taken from slightly different points of view. Passive stereo acquires both texture and depth, potentially in only a single shot from each camera. many dual and multi-view stereo methods exist, as summarized by Seitz et al. [2006], however in this thesis we will be mainly building upon the approach proposed by Beeler et al. [2011]. In this approach, a system to refine a coarse multi-view stereo reconstruction by minimizing an error function that integrates stereo, shading and smoothness terms is proposed. They exploit the small-scale strength of shape-from-shading methods[BZK85], and explicitly limit it to the spatial frequencies that are not captured by multi-view stereo. Furthermore, they track skin movement over time in a robust way, yielding a sequence of high resolution, high fidelity, fully corresponding dense facial surfaces.

Despite the high-resolution quality of the aforementioned reconstruction approaches, current methods are unable to capture one of the most important regions of the face - the eye region. In Chapter 3 we present the first method for detailed spatio-temporal reconstruction of eyelids. Tracking and reconstructing eyelids is extremely challenging, as this region exhibits very complex and unique skin deformation where skin is folded under itself while opening the eye. Furthermore, eyelids are often only partially visible and obstructed due to self-occlusion and eyelashes. Our approach is to combine a novel geometric deformation model with image data, leveraging multi-view stereo, optical flow, contour tracking and wrinkle detection from local skin appearance. Our deformation model serves as a prior that enables re-

construction of eyelids even under strong self-occlusions caused by rolling and folding skin as the eye opens and closes. The output is a person-specific, time-varying eyelid reconstruction with anatomically plausible deformations. Our high-resolution detailed eyelids couple naturally with current facial performance capture approaches. As a result, our method can largely increase the fidelity of facial capture and the creation of digital content.

1.2.2 Augmentation

For most practical applications, such as films, video games and medical simulations, the captured surfaces described above are insufficient. Different content is often required due to numerous reasons. The captured performers are typically restricted in motion due to the capture devices, which impairs the quality of their performance. Exaggerated or almost non-human expressions are often times more compelling than the realistic ones. Even the performance itself is usually altered, changed or completely re-generated in real time for games or post-production scenarios. For these reasons, tools to augment facial performances are required. These tools must be artistic friendly - fast, lightweight and provide powerful and yet intuitive control, while maintaining the original quality of the captured faces. To this end, a large variety of approaches have been pursued, which deform, augment or blend a set of captured facial expressions and sequences to generate new performances.

Such approaches include physically-based models, parametrization models, and motion capture. Physically-based models approximate the mechanical properties of the face such as skin layers, muscles, tissues, bones, etc., which are very computationally intensive, but have the potential to better include collisions and external stimuli [SNF05]. Parametrization models (or *rigs*) generate a facial pose as a (typically linear) combination of a number of facial expressions. By varying the weights of the combination, a wide range of facial expressions can be explored with little computation. Due to their intuitive controls, facial rigs based on blendshape models are particularly popular among artists for creating realistic looking facial animations [Lew+14]. Motion capture methods track features, usually identified by special makeup or markers, in space and over time and deform a captured expression to match the motion data [Gue+98].

In Chapter 4, we present a technique for adding fine-scale details and expressiveness to low-resolution art-directed facial performances, such as those created manually using a rig, via marker-based capture, by fitting a morphable model to a video, or through Kinect reconstruction using recent

faceshift technology. We employ a high-resolution facial performance capture system to acquire a representative performance of an individual in which he or she explores the full range of facial expressiveness. From the captured data, our system extracts an expressiveness model that encodes subtle spatial and temporal deformation details specific to that particular individual. Once this model has been built, these details can be transferred to low-resolution art-directed performances. We demonstrate results on various forms of input; after our enhancement, the resulting animations exhibit the same nuances and fine spatial details as the captured performance, with optional temporal enhancement to match the dynamics of the actor. Finally, we show that our technique outperforms the current state-of-the-art in example-based facial animation.

1.2.3 Display

As a last step of the facial content creation pipeline, the digital content is displayed to the viewer. While the challenges discussed in the previous sections are crucial to the perception of realism for human faces, equally important is the ability to display faces realistically. Over the last years, different technologies were developed in order to enhance the realism and feeling of immersion of digital content in general and digital facial content specifically. The classic way of displaying digital content is on a 2D display. This requires projecting the 3D content onto the 2D viewing plane, widely known as rendering. The rendering of faces has long been and still is an active field of research since human skin is particularly challenging [dLE07; Deb+00; JG10; DI11]. This difficulty has two origins. First, human skin exhibits multiple scattering effects, both on and inside the surface, which requires volumetric simulation. Second, the skin's reflectance varies both spatially and temporally. Through the contraction and dilatation of blood vessels the appearance of the skin varies as the face changes in expressions. To make things worse, the appearance of the face is also determined by a variety of small scale effects such as wrinkles, pores, and follicles.

At the same time, different technologies have been developed, aimed at increasing the sense of immersion. Display devices capable of conveying depth perception are the main ones. Such devices traditionally present offset images that are displayed separately to the left and right eye, and are then combined in the brain to give the perception of depth, according to the principle of stereopsis. These devices typically rely on anaglyph, polarization or active shutter methods, see [MB13] for a survey of advantages and disadvantages of the different methods. Auto-stereoscopic displays offer a wider range of visual cues to increase even more the sense of immersion [Ben+07].

Lastly, when attempting to render the humans with conventional or stereoscopic displays, non-verbal cues such as head pose, gaze direction, and facial expression are difficult to convey correctly to all viewers. Therefore, physical avatars have been used to give physical presence to a character, both in the field of entertainment and teleconferencing. Using a head-shaped display surface that intrinsically provides depth cues, simultaneously giving multiple observers their own perspectives. However, such physical avatars' movement and expressions are often limited due to mechanical constraints. In Chapter 5, we propose a complete process for augmenting physical avatars using projector-based illumination, significantly increasing their expressiveness. Given an input animation, the system decomposes the motion into low-frequency motion that can be physically reproduced by a given animatronic head and high-frequency details that are added using projected shading. At the core is a spatio-temporal optimization process that compresses the motion in gradient space, ensuring faithful motion replay while respecting the physical limitations of the system. We also propose a complete multi-camera and projection system, including a novel defocused projection and subsurface scattering compensation scheme. The result is a highly expressive physical avatar that features facial details and motion otherwise unattainable.

1.3 Publications

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

[Ber+13] Amit H. Bermano et al. "Augmenting Physical Avatars Using Projector-based Illumination". In: *ACM Trans. Graph.* 32.6 (2013), 189:1–189:10. ISSN: 0730-0301.

[Ber+14a] Amit H. Bermano et al. "Facial Performance Enhancement Using Dynamic Shape Space Analysis". In: *ACM Trans. Graph.* 33.2 (2014), 13:1–13:12. ISSN: 0730-0301.

[Ber+15] Amit H. Bermano et al. "Detailed Reconstruction of Eyelids". In: *to be published ACM Trans. Graph.* (2015).

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

[Ber+12] Amit Bermano et al. "Shadowpix: multiple images from self shadowing". In: *Computer Graphics Forum*. Vol. 31. 2pt3. Wiley Online Library. 2012, pp. 593–602.

[BVG11] Amit H. Bermano, Amir Vaxman, and Craig Gotsman. "Online reconstruction of 3d objects from arbitrary cross-sections". In: *ACM Transactions on Graphics (TOG)* 30.5 (2011), p. 113.

[Var+15] Orestis Vardoulis et al. "Single breath-hold 3D measurement of left atrial volume using compressed sensing cardiovascular magnetic resonance and a non-model-based reconstruction approach". In: *Journal of Cardiovascular Magnetic Resonance* 17.1 (2015), p. 47.

CHAPTER 2

Related Work

This thesis covers different fields in computer graphics and computer vision, including face and eyes capturing, performance modeling and transfer, and projection related issues. In order to summarize the works related to these subjects, we structured this chapter according to the facial content creation pipeline described in Section 1.2. Excluding projection related issues, which is relevant only to Chapter 5, all of the following works are related to some extent to all the projects that are discussed in this thesis (Chapters 3,4, and 5):

- Capture related methods are discussed in Section 2.1.
- Deformation, augmentation and modeling work is covered in Section 2.2.
- Projection related approaches are addressed in Section 2.3.

2.1 Capture

Our work depends heavily on 3D face capture. Chapters 4 and 5 employ existing methods and build upon them, while Chapter 3 aims to improve them. The latter is particularly related to other techniques that are tailored for reconstructing or modeling the eye region. In contrast to existing work, we present in Chapter 3 the first method specifically designed to capture person-specific eyelids, including the complex temporal behavior and self-folding that occurs during opening and closing of the eyes. Our work nat-

urally complements existing techniques for face and eye capture, forming a more complete and realistic digital face.

Face Capture. There has been a lot of work on capturing the geometry of human faces. Some methods focus on capturing high-resolution static poses, both active and passive [Wey+06; Ma+07; Bee+10; Gho+11], that can then be animated later with marker-based motion capture data [Wil90]. Others use space-time stereo to capture low-resolution 3D models at interactive rates [Bor+03; Wan+04a; Zha+04; ZH06]. Neither approach is capable of simultaneously capturing high-resolution spatial and temporal details. Bickel et al. [2007] combine high-resolution static geometry with motion capture data for large-scale deformations and add medium-scale expression wrinkles tracked in video. Huang et al. [2011] leverage motion capture and static 3D scanning for facial performance acquisition.

Recent approaches use high-speed cameras and photometric stereo to capture performance geometry [Wen+05; Jon+06; Ma+08]. Some of these techniques use time-multiplexed illumination patterns and consequently require an acquisition rate that is a multiple of the final capture rate. Wilson et al. [2010] introduce a temporal upsampling method to propagate dense stereo correspondences between frames to reconstruct high-resolution geometry for every captured frame. Bradley et al. [2010] use a completely passive system with high-resolution cameras.

Reduced hardware approaches such as binocular [Val+12] and monocular [Gar+13; SKSS14; Shi+14] facial capture methods often use shape from shading approaches [Wu+11] to recover fine-scale facial details. Real-time facial animation methods sacrifice quality in favor of reconstruction speed, typically through the use of generic facial priors [Wei+11; Rhe+11; Cao+13; BWP13; Li+13; CHZ14]. These techniques are able to animate generic shapes that are part of the shape priors, but are unable to recover person-specific details.

In Chapters 3 and 4, We use an extended version of the passive system by Beeler et al. [2011] to capture dynamic high-resolution 3D geometry for our expression database (Chapter 4), and as bases for our eyelids reconstruction (Chapter 3). The specifics of how the initial capture is performed in both cases are not important, and a number of alternative methods could be used as well, such as [Hua+11b; KH12].

Eyes and Eyelids. The important role that eyes play in computer graphics applications has led to a number of research topics including eye motion and

blink animation, iris and eyelid modeling, and high-quality eye capture. A detailed survey of eye and gaze animation methods was recently presented by Ruhland et al. [2014]. In the following we summarize the most related methods.

Since the seminal work of Lee et al. [2002] on keeping a character's eyes "alive", several models of eye motion and blinks have been derived from motion capture and video data [DLN05b; WLO10; Tru+11; LMD12]. However, these approaches do not focus on detailed eyelid reconstructions that would provide person-specific eyelid wrinkle formation.

On the topic of reconstruction, François et al. [2009] estimate the multilayered shape and approximate scattering properties of the iris from a single camera image. More complete and detailed reconstructions were recently shown by Bérard et al. [2014], who use a multi-camera and multi-light setup to capture all the visible components of the eye in very high-resolution. Note that these methods are complementary to ours since they focus exclusively on the eyes and we are considered with the skin surrounding it.

Our work is not the first to model eyelids. A method for modeling the eye region has been presented by Sagar et al. [1994] in the context of surgical simulation. The eyelids are modeled by a simple NURBS surface that is fit to a single face scan and then manipulated by hand. In contrast, we reconstruct temporally-varying high-resolution geometry, capturing the unique wrinkling behavior of each individual eyelid.

2.2 Augmentation

Augmentation issues are discussed mainly in Chapter 4, covering facial animation, and performance transfer and synthesis. However, at a higher level, the proposed method in Chapter 3 is akin to methods that model or generate wrinkle geometry for faces and clothing, and Chapter 5 address in part performance transfer as well. In the following we discuss previous work in these areas.

Wrinkle Modeling. One of the defining characteristics of eyelids is their natural self-folding behavior, which can be considered an extreme case of wrinkling. Bickel et al. [2007] decompose the face into multiple scales in order to enhance low-resolution marker-based motion capture with fine-scale wrinkles captured on the forehead and under the eyes, and then learn the correspondence of skin-strain to wrinkle formation for real-time editing and wrinkle transfer to virtual characters [Bic+08a]. Dynamic wrinkles are also

captured by Ma et al. [Ma+08] and modeled as a polynomial displacement map on top of a low-resolution face model. This approach of modeling wrinkles as a displacement map layer is very common in facial animation [K̈02; DMB11; Li+15]. Skin wrinkles can also be generated through physically-based simulation of the face if the anatomy is sufficiently modeled [MT+02; ZST05; WM14]. Other techniques to enhance low-resolution facial performances with high-resolution wrinkles include mapping expressions into a common shape space and transferring the high-frequency details [Ber+14b], or manually specifying wrinkle curves with an artistic tool [BKN02; LC04]. While these methods target wrinkle modeling on the face, they do not address the complex wrinkling behavior of eyelids.

For clothing, a similar trend has emerged to enhance low-resolution cloth models with previously-generated fine-scale wrinkles [MC10; FYK10; Wan+10; Roh+10; SSH12; Kim+13; ZBO13]. These methods typically assume a coarse cloth simulation is available, which provides the underlying motion of the surface, and thus are not well-suited for capture scenarios. On the other hand, Popa et al. [2009] procedurally generate wrinkles for captured garments. Starting with a reconstruction of the low-frequency garment shape [Bra+08], they introduce temporally coherent high-frequency wrinkles that correspond to detected edges in the capture images. Such an approach could also improve the shape of wrinkles on a captured face, however the method does not consider wrinkles with strong self-occlusions such as eyelids.

Facial Animation. Facial animation has a long history that goes back to the early '70s [Par74]. Some methods use models of facial anatomy [Wat87; TW93] that can be combined with physical models of skin deformation [WKMT96; SNF05; VLR05; ZS05]. Another approach is to use deformable 3D face models [Bla+03; Vla+05] and fitting them to video data [LRF93; Ess+96; DM96; PSS99; Dal+11]. Methods based on example poses and shape interpolation (i.e., blendshapes) [LCF00; CB02; Lew+05; Seo+11] are especially popular in the entertainment industry because of their intuitive and flexible controls and can even be driven in real-time from video [CXH03] or the Microsoft Kinect device [Wei+11]. Similar concepts can also be applied to drive a set of hand-drawn faces for generating performance-driven, "hand-drawn" animation in real-time [Buc+00]. None of these approaches reach the quality of high-resolution performance-driven facial animation from person-specific captured data, and animation of subtle facial details and dynamics are still elusive. Our approach in Chapter 4 tries

to bridge the gap between traditional facial animation and high-quality 3D face scanning.

Deformation and Detail Transfer. Static geometry can be enhanced with details transferred from different models by means of simple displacements (for small details) or differential coordinates (for substantial enhancements) [Sor+04; Tak+11]; With such methods, the transferred detail is explicitly given, rather than being a function of the low-resolution pose. Deformation transfer techniques [SP04] such as expression cloning [NN01; Pyu+03] transfer vertex displacements or deformation gradients from a source face model to a target face model with possibly different geometry. Similarly, data-driven approaches (e.g., based on Canonical Correlation Analysis [FKY08] or Gaussian Process models [MLD09]) learn and transfer facial styles. These techniques are typically applied to low-resolution geometry or low-frequency deformations. Golovinskiy et al. [2006] add static pore detail from a database of high-resolution face scans using texture synthesis. Huang et al. [2011] train a collection of mappings defined over regions locally in both the geometry and the pose space for detailed hand animation. Bickel et al. [2008] use radial basis functions to interpolate medium-scale wrinkles during facial performance synthesis and transfer. Ma et al. [2008] add high-resolution facial details to a new performance using a compressed representation of vertex displacements. Notably, Alexander et al. [2010] use high-resolution scans to generate a detailed blendshape rig. In contrast to these methods, in Chapters 4 and 5 we present frameworks that enable both spatial and temporal performance enhancement and transfer, which can be applied to various forms of art-directed facial animation, augmenting the high-resolution details and matching the dynamics of the particular target face.

Temporal Performance Synthesis. The temporal aspects of facial performances are very important for synthesis of new facial animations from speech [BCS97; Bra99; EGP02; KT03; Cao+04; Ma+04; DLN05a]. Most of these approaches record facial motion of speaking subjects and then recombine the recorded facial motion from learned parametric models to synthesize new facial motion. Chai and Hodgins [2007] learn a statistical dynamic model from motion capture data and generate animations from user-defined constraints solving a trajectory optimization problem. However, none of these methods take high-resolution spatial details into account. Instead of learning a model, in Chapters 4 and 5 we use a simpler and more general data-driven approach for performance synthesis of temporal and spa-

tial details. Note that we do not specifically target temporal enhancement of speech animation.

Performance Transfer. Acquiring the expression of real faces and applying them to computer-generated models is a central component for creating lifelike performances [HPL06]. A common method of performance transfer is encoding facial motion as a linear combination of target shapes and transferring the weights. The basis shapes can represent facial action units based on the facial action coding system [EF77] or learned from data [CB02; BV99; LWP10]. For non-rigid mapping of the source performance to the target model, alternatively the deformation field can be directly transferred by establishing dense correspondence [NN01; LSP08]. Common to these remapping techniques is that there is a static mapping between source and target expression. However, as the motion gamut of the animatronic head is very limited, we desire a dynamic, temporally local compression. Inspired by the observation that the source and target movements should be similar, Seol et al. [2012] present a space-time facial animation re-targeting approach, interpreting movement as derivative in time and formulating the re-targeting problem as a Poisson equation. In our setup, we represent the motion of the robotic head in the constraint space of control parameters of the head [Bic+12] and compute dense correspondences for transferring facial details and spatio-temporal optimization of the animatronics' head motion. In contrast to Seol et al. [2012], we do not optimize for global blend shape weights, but instead re-time the constrained coarse motion of the animatronic head to match the input motion (Chapter 5) or input key-frames to database sequences (Chapter 4).

2.3 Display

In Chapter 5, we propose a method to display digital facial content or animatronics heads using projection. In this section we discuss previous solutions proposed to solve problems that concern this matter.

Projected Avatars. Animated humanoid robots, called animatronics, are an old field of research. While currently high-quality animatronics exist that have a quite natural appearance [Ish06; NIH07a; Bic+12], their movements and expressiveness are still limited and thus, while they appear almost like real humans, they are positioned in the deep dip of the uncanny

valley. To overcome this problem, several research groups tried to use spatially varying illumination provided by projectors to superimpose the humanoid's face with dynamic textures to give it a more natural and dynamic appearance [Lin+09; MEB12; MIR12; Kur+11]. First described as a generic principle in [Ras+01], these approaches use a uniform white generic face geometry and apply projective texture mapping to superimpose colors and texture. While some of the heads are able to move rigidly, they are still significantly limited in their physical motion range. In contrast to those, in Chapter 5 we project onto a dynamic animatronic head having flexible, pigmented silicone skin. Thus, the physical head alone already enables limited non-rigid movement and the projection is employed for adding detailed shading on top.

Light-Transport-Based Projection Image Compensation Algorithms. The usage of projectors to change or enhance surface appearance has been an active research area for more than one decade. In [WB07], a light-transport-based radiometric compensation method is described that extends local methods (cf. [Bim+07] for an overview) to compensate for global illumination effects, such as defocus, refractions, diffuse, and subsurface-scattering for a particular camera view within the bounds of the capabilities of the used projector and camera. A quad-tree-based light transport measurement [Sen+05] is used, which takes up to several hours and thus is not practical for dynamic surfaces. This idea was recently extended in [Law+11; Ali+12] to generate a high-quality multi-projector compensation. Projector-camera systems can also be used to directly estimate the defocus of the projected pixels. If this defocus is measured and modeled as a point spread function (PSF) between the projector and the camera, an adapted image can be calculated which, up to a certain extent, compensates the projection defocus to make the image appear less blurred. Several approaches use a camera to evaluate the defocus on the surface and apply image filtering to generate a compensation image [ZN06; OS08]. Multiple projectors were used in [BE06] to generate a blended projection with pixel contributions from several projectors to minimize the defocus. As this approach requires overlapping projections, the overall contrast of the system is reduced because of summed black intensity contributions. In [Gro+10], a programmable aperture was integrated into the projector to generate a content-optimized image deconvolution. All compensation approaches so far, however, correct the defocus for a specific camera view and don't consider the oblique blur that is generated by a surface point that is not parallel to the camera's image plane. This effect was considered in [NIS11], which uses multiple projectors together with known geometry to calculate the camera-independent pixel contribu-

tion. Our method applies a camera-independent compensation differently: by analyzing and modeling the system's defocus properties for a specific projection volume independent of the dynamic projection surface. These precomputed parameters are looked up for the given surface geometry and, in combination with a description of its subsurface scattering, are used to globally optimize the projection images for all used projectors.

CHAPTER 3

Eyelids Reconstruction

The human face is the most important part of a person for conveying identity and emotion and therefore of central interest when creating realistic digital humans for computer games and films (Section 1.1). For identifying emotions, humans mainly use a consistent selective sampling of visual information from the eye region and, to a lesser extent, the mouth region [Smi+05; PE12]. Subtle details such as the twitch of an eyelid and the formation of small wrinkles significantly contribute to the realism of human faces and the perception of emotions. However, despite the important role of the eye region, existing capture technology is usually not able to provide an adequate level of geometric detail and motion to reproduce these subtleties. In practice, achieving realistic eyelid motions and skin deformation of the sur-

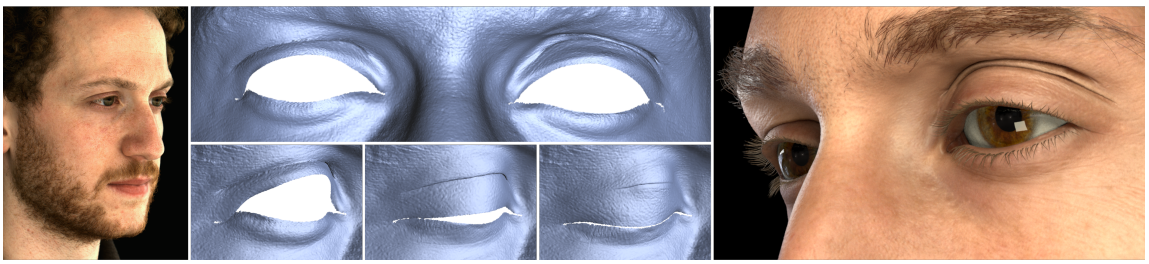


Figure 3.1: *Our method extends high-resolution facial performance capture with a reconstruction approach that targets eyelids. We produce detailed, spatio-temporal eyelid reconstructions, even during complex deformation and folding that occur in the eye region. The result can be used to create high-fidelity digital doubles, as shown on the right.*

rounding area requires significant manual modeling efforts by highly skilled artists.

Acquiring this region is extremely challenging due to several reasons. In an expressive performance, eyelids undergo extreme deformations and wrinkling. The skin rolls and folds inward when the eye opens, and stretches over the eyeball when the eye is shut. Due to concavities and eyelashes, there is significant self-shadowing, inter-reflections, and partial occlusions. Even worse, in many facial expressions a significant part of the eyelid is folded in and not visible at all. We desire an accurate performance capture that delivers consistent geometry in correspondence over time whenever visible, and deforms non-visible parts in a plausible way. Unfortunately, existing dense performance capture approaches cannot handle these extreme deformations and occlusions.

In this Chapter, we aim to improve the first step in any facial content process - capturing (Section 1.2), by addressing this problem and introducing a novel reconstruction and tracking scheme that combines a geometric deformation model with image-based data. The model is motivated by the physiological behavior of the eye - skin interface and constrains the reconstruction to anatomically plausible motions. This prior is required due to noise and missing data in the depth information, inherently caused by eyelashes and self-occlusions. In addition, we observe that wrinkles greatly change the local appearance of the skin, and thus their location can be accurately determined from images and local motion. This combination of anatomically motivated priors, depth information, and image-based data, makes detailed eyelid reconstruction feasible.

Such an approach has several advantages. We obtain a single, consistent mesh over time, which allows our result to be directly combined with existing facial performance capture approaches, or to be used to create a data-driven blendshape rig. Our method provides plausible deformations even for regions that are occluded. We are able to capture the dynamic effects of eyelids, which is important because the location and shape of wrinkles is not only dependent on the current state of the eye region but also its history, a phenomenon referred to as hysteresis. As our method is agnostic to the capture approach, it can be easily integrated into any performance capture pipeline, be it passive or active, that records sufficiently high-resolution footage of the eye region. As we demonstrate with several results, our system allows the reconstruction of an expressive, dynamic model of the eye region at a quality level that has never before been possible, increasing the fidelity of this very important facial component in the creation of digital doubles.

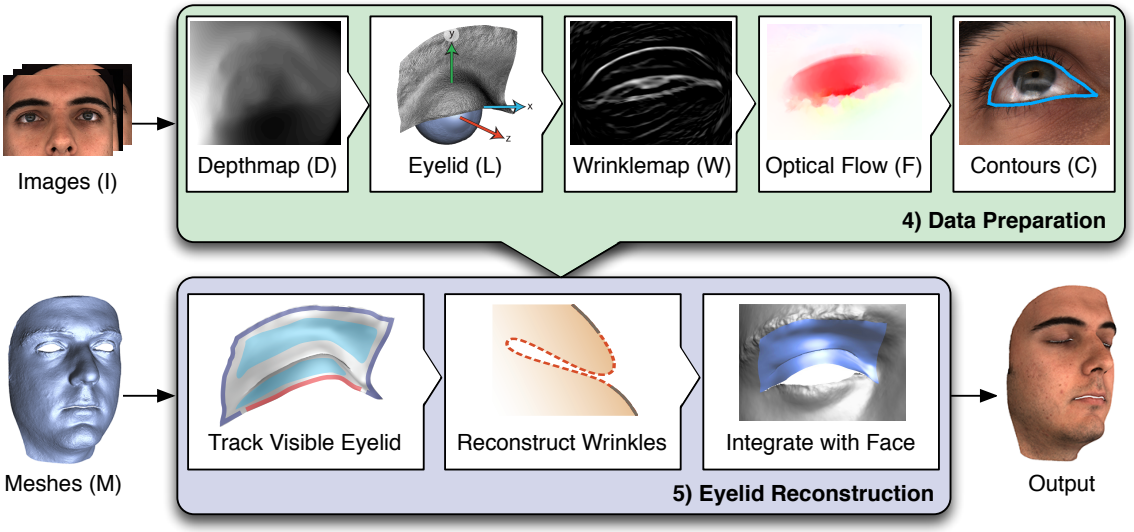


Figure 3.2: *Overview of the system. Starting from passively acquired image data, we compute for each frame depth information, optical flow, track the contours of the eye-skin interface, and the probability where wrinkles are forming (Section 3.2). Based on this data, our tailored deformation model deforms the eyelids over time, accurately tracking the actors performance (Section 3.3). Finally, the eyelid meshes are integrated with the reconstructed facial performance to provide a complete face model.*

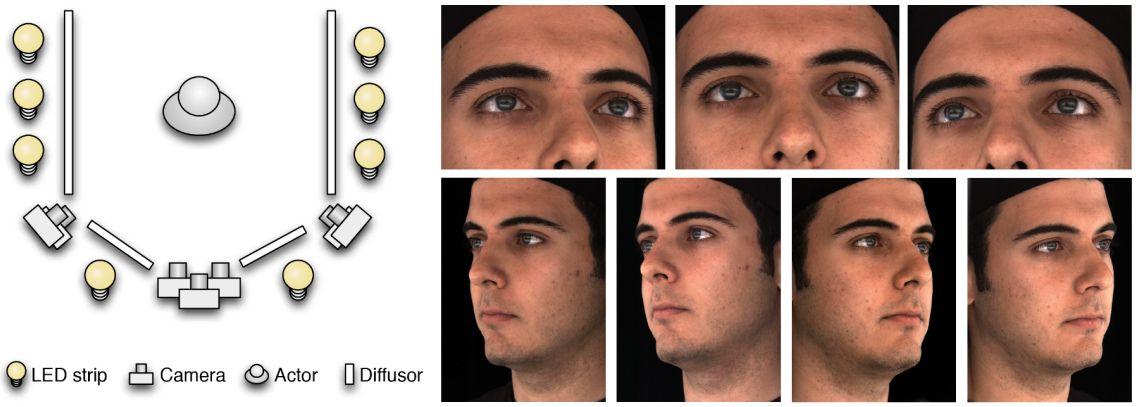


Figure 3.3: *The setup (left) consists of seven cameras, where the three central ones are zoomed in. LED strips mounted around the actor and diffused by frosted paper provide a flat illumination. An exemplary dataset for one frame is shown on the right.*

3.1 Method Overview

Our system, schematically depicted in Figure 3.2, starts by capturing a performance of the eyes using off-the-shelf cameras, as described in Section 3.2.1. The images are then analyzed to remove eyelashes and generate a spatiotemporal reconstruction of the face shape along with per frame depths maps (Section 3.2.2). Optical flow computed frame by frame is misled by the skin wrinkling and needs to be corrected (Section 3.2.5) using wrinkle probability maps which indicate where wrinkles are most likely to form (Section 3.2.4). Finally, we also track accurate eyelid contours over time (Section 3.2.6) to ensure faithful reconstruction of the visually important interface between the eyelid and the eye.

For all four eyelids, we manually create template meshes (Section 3.2.3) which will be continuously deformed from frame to frame. As shown in the lower part of Figure 3.2, eyelid tracking starts by deforming the visible part of the eyelid using constraints from optical flow, tracked eyelid contours and the surrounding face mesh (Section 3.3.1). We then reconstruct the parts of the eyelid which were subject to wrinkling and are thus not visible. The reconstruction produces plausible wrinkles that are visually pleasing (Section 3.3.2). Lastly, we use the eyelid templates as control meshes to deform the face mesh (Section 3.3.3) resulting in a complete facial performance with accurately tracked eyelids as shown in the bottom right corner of Figure 3.2.

3.2 Data Preparation

In this section we describe how to generate and prepare the input data required for eyelid reconstruction.

3.2.1 Data Acquisition

The image data is acquired using a multi-view setup consisting of seven synchronized video cameras, each providing roughly 40 frames per second at about 1MP. Three cameras are zoomed in to get higher resolution on the eye region and the other four are split in pairs of two to capture the full face. As illumination, we mount LED strips on a cage around the actor and diffuse them with frosted paper. Figure 3.3 shows the setup on the left and the images captured at one point in time on the right.

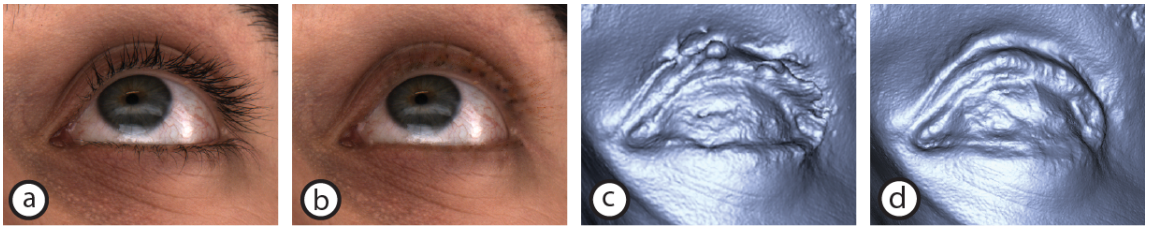


Figure 3.4: *Eyelashes pose problems for the reconstruction as they occlude the underlying skin (a) and confuse stereo methods, leading to noisy geometry (c). We adopt the inpainting approach proposed by Beeler et al. [2012a] to remove the eyelashes (b), which improves the reconstructed geometry substantially (d).*

3.2.2 Face Mesh Reconstruction

From the acquired images, we reconstruct the spatio-temporal shape of the face using the method of Beeler et al. [2011], which provides high-resolution per-frame tracked meshes in dense correspondence. In addition to the tracked meshes, we also compute per-frame depth maps D using Beeler et al. [2010], which contain information in areas not covered by the tracked meshes, such as the eyes. One major problem for stereo-based reconstruction methods are the eyelashes, which occlude the underlying skin and confuse stereo matching, causing considerable artifacts in the reconstructed depth maps. Therefore we adopt the inpainting approach proposed by Beeler et al. [2012] to remove the eyelashes from the input images before reconstruction, which greatly improves the reconstruction quality as can be seen in Figure 3.4.

3.2.3 Eyelid Initialization

We manually create a template mesh for each of the four eyelids (upper and lower, left and right) once per actor. Per Section 3.3, the template meshes should consist of a regular grid of vertices $\mathbf{v}_{i,j}$, in which the rows i are aligned with the predominant wrinkle orientation and the columns j run orthogonal across the wrinkles. This structure allows to efficiently process the eyelid area to detect and reconstruct wrinkles (Section 3.3.2). One way to create the template meshes would be to model them in 3D. This, however, would require the user to be familiar with 3D modelling. Instead we propose a simpler means to generate the template meshes by drawing a few curves on a closed-eye image of the actor (see Figure 3.5, left). From these curves a 2D regular grid is created (center) and lifted to 3D using the computed depth maps D to generate the eyelid (right). In our experiments, we have traced 10

curves for each eye, loosely approximating wrinkle flow lines, from which a grid of size 120×100 vertices is generated. We found that the best expression to use is one where the eyes are closed and the eyebrows are raised, since the entire eyelid is visible and the skin is least compressed. We will refer to this as the rest pose later on.

The user also initializes a reference coordinate frame, approximately in the center of the eye socket, with the z-axis pointing forward and the y-axis up (Figure 3.5, right). This coordinate frame follows the rigid head motion computed by rigid stabilization [BB14] and is used both to reconstruct the eyelid wrinkles (Section 3.3.2) and to compute the wrinkle probability map described in the next section.

3.2.4 Wrinkle Probability Map

The wrinkle probability map encodes the likelihood that a pixel is part of a wrinkle, and is computed for each frame from the inpainted and histogram normalized images using oriented kernels. Specifically we employ anisotropic difference of Gaussians $\mathcal{N}(\sigma_x, \sigma_y, \theta) - \mathcal{N}(\sigma_x)$ for seven different orientations θ in the range of $\pm 20^\circ$, where we set $\sigma_x = 8$ and $\sigma_y = 0.1\sigma_x$, and record the maximum response in the wrinkle map. Other oriented kernels, such as Gabor, could also be applied. While this identifies wrinkles it also captures a lot of noise caused by areas of similar appearance. To improve the signal-to-noise ratio we propose the following three steps. First, since the wrinkles we are interested in tend to form concentrically around the center of the eyes in the images, we rotate the oriented kernel based on the relative position to the closest eye center. Second, we employ spatio-temporal hysteresis [Can86], which keeps only pixels whose probability is either higher than a given threshold ξ_u, or which are connected to such pixels in space or time via some other pixels with probabilities no lower than ξ_l. We use $\xi_u = 0.05$, $\xi_l = 0.01$ for all results. Third, since the inpainting might have missed a few eyelashes, which can happen if they cluster, we consolidate wrinkle maps from multiple views and filter wrinkle probabilities where the views do not agree.

3.2.5 Optical Flow

To be able to track the eyelid over time, we compute optical flow F from one frame to the next using the method of Brox et al. [2004] on the inpainted images. A source-sink map S encodes the density of the optical flow and is computed by accumulating the inbound flow vectors for every pixel. Areas

where the flow vectors diverge are considered a source and appear dark in the visualization, and areas where they converge are considered a sink and they appear bright (see Figure 3.6). While generally very reliable, optical flow performs poorly at the wrinkles. Despite the motion of the eyelid surface during wrinkle formation, the appearance around the wrinkle remains similar due to shading, and this can confuse the flow computation. The incorrect flow vectors become very apparent when inspecting the source-sink map (Figure 3.6, top row). As we can see, the flow compresses on both sides of the wrinkle and not inside of it, which would be the correct sink.

We devise a method to correct the optical flow. Using guidance from the source-sink map S, the wrinkle probability map W (Section 3.2.4) is diffused smoothly to spread out the probabilities. As we do not intend to reduce existing probability but just spread it out, we iteratively update the probability map using

$$\hat{W}^{k+1} = \max\left(\hat{W}^k, \mathcal{N}\left(\hat{W}^k\right)\right), \tag{3.1}$$

where $\hat{W}^k$ denotes the diffused wrinkle probability map at the k-th iteration, $\hat{W}^0 = W$ and $\mathcal{N}$ is a Gaussian filter of size 7×7 (and an eye region is approximately 400 pixels wide in the image). The number of iterations required is determined by the distance of the sink to the true wrinkle location, which we found to be consistent in our examples and thus the same number of iterations (30) were applied to all frames, leading to a smoothed map as shown in Figure 3.7.c. The gradient of this map $\nabla\hat{W}$ (Figure 3.7.d) indicates the direction towards the closest wrinkle and will be used to correct the flow in a two-step process. First, we diffuse the source-sink map S (Figure 3.7.e) towards the wrinkle center, as this will determine the area in which the flow needs to be corrected. To do so we employ a variant of anisotropic diffusion [PM90]:

$$\hat{S}^{k+1} = \hat{S}^k + \lambda\psi\left(c\left(\nabla\hat{W}\right)\nabla\hat{S}^k\right), \tag{3.2}$$

where $\hat{S}^k$ denotes the diffused source-sink map at the k-th iteration and $\hat{S}^0 = S$. Instead of preventing smoothing along the gradient as was the goal in Perona and Malik [1990] we control the diffusion to spread predominately in the positive direction of the gradient. As diffusion coefficient c, we therefore choose

$$c\left(\nabla\hat{W}\right) = \left|e^{-\left(\frac{\nabla\hat{W}}{\kappa}+1\right)}\right|_{0,1}, \tag{3.3}$$

where κ (0.01) controls the sensitivity and $|\cdot|_{0,1}$ clamps to the range of $[0, 1]$ to warrant the maximum principle. The retaining function $\psi(x)$ attenuates the decay by multiplying x with a user given parameter (0.1) whenever $x < 0$, thus spreading this information to a larger region. The timestep λ was set to $1/8$ and the diffusion is run for 60 iterations leading to the result shown in Figure 3.7.f.

We employ $\hat{S}$ to attenuate diffusion of the flow field outside of the wrinkle neighborhood by including it in the diffusion coefficient as

$$c\left(\hat{S}, \nabla\hat{W}\right) = \left|\hat{S}e^{-\left(\frac{\nabla\hat{W}}{\kappa}+1\right)}\right|_{0,1} \tag{3.4}$$

and then diffuse the flow field F using Equation 3.2. To prevent flow vectors from overshooting the wrinkle location, we only update them if the wrinkle probability gradients at the origin and destination of the flow vector point in the same direction, i.e. the flow remains on the same side of the wrinkle. Figure 3.6 shows how the original flow and source-sink map (top row) are corrected by this approach (bottom row).

3.2.6 Eyelid Contours

The time-varying 2D eyelid contours are invaluable constraints for reconstructing accurate eyelid deformation. For this reason we also pre-compute contour curves for each frame. The contours are tracked in image space from a single front view using a two step method. First, we compute an initial contour shape estimate using the regression framework proposed by Cao et al. [2012]. This framework has shown to work well on related problems [Cao+13; CHZ14] but any similar system, such as active appearance trackers [CET01], may be employed. We then refine the contour position in image space using optical flow.

For each actor we choose a small set of frames in which we manually trace the eyelid contour (Figure 3.8.a). Each of these reference contours is represented by a set of landmarks, placed equidistantly along the contour from the inner to the outer eye corner. From these samples we then train an eye-specific contour tracker. In our experiments, we used a reference set of $20 - 30$ contours, and 20 landmarks. For convenience, the reference set was constructed by starting with an initial blink sequence, and was then iteratively expanded by adding frames that caused tracking failures (up to at most 3 iterations).

To track the contours over the sequence, we apply the contour tracker on the frames taken from the same view. Each frame's tracking result is used to initialize tracking in the next frame. The tracking results provide a good initial estimate of the contour shape and position (Figure 3.8.b), but are not sufficient to accurately constrain the eyelid reconstruction, and thus need to be further refined. For each frame we retrieve the most similar reference frame by comparing the shape of the predicted contour to the reference contours. We then compute optical flow [Bro+04] between the reference image and the current image and use the flow vectors to deform the reference contour into the current frame yielding subpixel-accurate registration to the reference frame (Figure 3.8.c).

However, sequential frames may be matched to different reference frames, which could lead to temporal jitter since the reference contours themselves exhibit some inaccuracies as they are hand-drawn. We thus smooth the contours temporally over the entire sequence using optical flow computed between frames to produce accurate, temporally smooth eyelid contours.

3.3 Eyelid Reconstruction

In this section, we describe our eyelid deformation model and our method for robust eyelid reconstruction. Our goal is to evolve the eyelid created in Section 3.2.3 over time t. The eyelid is represented by a template mesh L, which consists of regularly sampled vertices $\mathbf{v}_{i,j}$ along the horizontal direction j corresponding to the dominant main wrinkle orientation and the orthogonal vertical direction i, as explained in Section 3.2.3 and illustrated in Figure 3.5. Our deformation model follows a two-step process: first, as described in detail below, we deform the skin based on optical flow and tracked contour data. This provides the desired behavior in areas visible both in the current and previous frames and undergoing moderate deformation, but optical flow is unreliable in more challenging cases. Even though the flow correction described in Section 3.2.5 improves the reliability in regions visible in both frames, it is unable to guide the deformation of the occluded mesh regions. Thus, for newly occluded regions, vertices will be compressed at the wrinkle location (see Figure 3.9 (left)). To address this challenge we identify wrinkle regions and propose a dedicated wrinkle model that is parameterized with a small set of distinctive feature points. These feature points can be efficiently estimated from the acquired data and allow plausible reconstruction even in regions with extreme skin deformations.

3.3.1 Visible Skin Deformation

The first step of our deformation model is driven by the visible areas of the skin - we deform the eyelid using optical flow where it can be trusted, while respecting tracked boundary conditions from the surrounding face. The extreme deformations occurring around the wrinkle areas are handled in a second step, described in Section 3.3.2.

Close inspection of the eyelid reveals that the eye-eyelid interface transforms mostly rigidly, as it fits tightly around the eye shape while sliding over it. Consequently, this area preserves its shape based on the underlying eye and mostly just rotates when the lid opens – unlike the rest of the eyelid, which undergoes strong wrinkling. To reflect this we combine two linear thin-shell energies [BS08] to deform the eyelid (E_S, E_I), guided by three different data terms (E_C, E_B, E_F). Figure 3.10 illustrates the spatial distribution of these energies on the mesh. The first energy regularizes the deformation based on the shape of the lid at the previous frame L^{t-1} and is given as

$$E_S = \sum_V \left\| \Delta_{L^{t-1}} (\mathbf{v}_{i,j}^t - \mathbf{v}_{i,j}^{t-1}) \right\|^2, \tag{3.5}$$

where $\Delta_{L^{t-1}}$ is the discrete Laplace-Beltrami operator for the eyelid mesh L^{t-1} and V denotes all vertices of the mesh. The second energy reflects the deformation driven by the rigidly transforming region at the eye-eyelid interface V^I. It seeks to deform this region to match the rest pose, up to a global rotation R^t:

$$E_I = \sum_{V^I} \left\| \Delta_{L^0} (R^t \mathbf{v}_{i,j}^t - \mathbf{v}_{i,j}^0) \right\|^2, \tag{3.6}$$

where Δ_{L^0} is again the discrete Laplace-Beltrami operator, this time computed from the rest pose. Note that the two Laplace-Beltrami operators only differ in the cotangent weights, which is required to account for skin compression while the lid opens. Figure 3.10.a depicts the regions regulated by these energies.

The eye-eyelid interface itself does not transform purely rigidly, but undergoes some deformation due the shape of the underlying eye. We account for this by incorporating the contours computed in Section 3.2.6 as an energy term

$$E_C = \sum_{V^C} \| P(\mathbf{v}_{i,j}^t), C^t \|_\ell^2, \tag{3.7}$$

where $\|\cdot,\cdot\|_\ell$ denotes the point-line distance in image space computed by projecting $\mathbf{v}_{i,j}^t$ into the camera image using the camera projection matrix P. The contour imposes constraints on the vertices V^C at the eye-eyelid interface of the eyelid mesh L. The remaining boundary should deform such that it is compatible with the face mesh M to alleviate integration later on (Section 3.3.3). For the vertices V^B in the outer two rings at these boundaries, we wish to constrain the motion to be similar to the motion of the corresponding points C^B on the face mesh. A correspondence $\mathbf{c}_{i,j}^B \in C^B$ is computed as the closest point in rest pose to the eyelid vertex $\mathbf{v}_{i,j}^0 \in V^B$. Encoding the correspondence in barycentric coordinates of the adjacent triangle allows to propagate it in time consistently with the face mesh. The boundary energy term is then

$$E_B = \sum_{V^B} \left\| \mathbf{v}_{i,j}^t - \mathbf{c}_{i,j}^{B,t} \right\|^2, \tag{3.8}$$

The valid vertices in the interior of the eyelid $\mathbf{v}_{i,j}^t \in V^{F,t}$ at time t are constrained by optical flow. We compute positional constraints $\mathbf{c}_{i,j}^{F,t} \in C^{F,t}$ for these vertices by projecting them into the main camera's image plane, advecting them using the optical flow F and elevating them back into 3D using the depth maps D. A vertex is considered to be valid if it is (1) visible from the main camera, (2) does not exhibit a high enough wrinkle probability, and (3) is sufficiently far from the boundary (we use a 5-ring margin from the border in all our experiments). The flow energy term

$$E_F = \sum_{V^{F,t}} \gamma_{i,j} \left\| \mathbf{v}_{i,j}^t - \mathbf{c}_{i,j}^{F,t} \right\|^2 \tag{3.9}$$

where $\gamma_{i,j}$ is a confidence weight indicating how much the constraint can be trusted. The confidence is provided by the multi-view geometry reconstruction method and is a measure of how similar the neighborhood of this vertex looks in the different views. This helps overcome outliers caused, for example, by occluding eyelashes. Note that the vertex set $V^{F,t}$ associated with E_F may change over time. The vertex sets associated with all other energy terms remain constant throughout the sequence. Figure 3.10.b illustrates the mesh regions contributing to each data term.

Combining the individual terms together yields the total energy

$$E = \lambda_F E_F + \lambda_B E_B + \lambda_C E_C + E_I + E_S, \tag{3.10}$$

where $\lambda_{F,B,C}$ are user parameters. In our experiments we fixed $\lambda_F = \lambda_B = 30$ and $\lambda_C = 300$.

Unfortunately, the rotation R^t used in E_I (Equation 3.6) is also unknown and needs to be estimated as well. Following Sorkine and Alexa [2007] we interleave estimation of deformation and rotation and iterate both of them three times, starting with estimating the deformation.

3.3.2 Wrinkle Reconstruction

After deforming the visible parts of the eyelid, we process the hidden and newly occluded areas. For these areas flow computation is not possible and the best the flow correction (Section 3.2.5) can do is to compress the problematic vertices into the wrinkle area. Figure 3.9.b depicts schematically how the vertices from Figure 3.9.a are aggregated at the wrinkle location. The problem is further aggravated since the multi-view stereo method cannot accurately reconstruct small scale details (Figure 3.9. (right)) and as a consequence the geometry in the wrinkle area cannot be trusted. In this section we describe how we create an anatomically plausible wrinkle shape and move these vertices into the wrinkle valley in a physically meaningful manner (Figure 3.9.c).

We start by projecting the wrinkle probability map W^t onto the eyelid L^t and assign a wrinkle probability $w^t_{i,j}$ to every vertex. The eyelid mesh has been designed such that it allows to efficiently identify distinctive wrinkle feature points from which we can construct the hidden part of the wrinkle. Figure 3.11 illustrates the extracted feature points schematically. Traversing a wrinkle cross-section from top to bottom will sequentially produce the *top of wrinkle* ($\mathbf{v}_{top}$), the *front-buckle* ($\mathbf{p}_1$), the *back-buckle top* ($\mathbf{p}_2$), the *back-buckle bottom* ($\mathbf{p}_3$) and the *bottom of wrinkle* ($\mathbf{v}_{bottom}$). Note that $\mathbf{v}_{top}$ and $\mathbf{v}_{bottom}$ correspond to actual vertices of the mesh, where $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are points in space. The individual feature points are computed as follows:

1. *Top of wrinkle* ($\mathbf{v}_{top}$): First vertex with wrinkle probability $w^t_{i,j} > \xi$ when traversing the vertices $\mathbf{v}^t_{i,j}$ of a column j from top to bottom. We set $\xi = 0.1$ for all our results.
2. *Bottom of wrinkle* ($\mathbf{v}_{bottom}$): Last vertex with wrinkle probability $w^t_{i,j} > \xi$. The vertices between $\mathbf{v}_{top}$ and $\mathbf{v}_{bottom}$ denote the wrinkle segment ($\mathbf{v}_{top}$ to $\mathbf{v}_{bottom}$) (Figure 3.11.a).
3. *Front-buckle* ($\mathbf{p}_1$): Computed by projecting the weighted average $\hat{\mathbf{p}}_1$ of all visible vertex positions in the wrinkle segment onto the ex-

tended plane from $\mathbf{v}_{top}$ (Figure 3.11.b). The weighted average is computed as $\hat{\mathbf{p}}_1 = \sum w^t_{i,j}\mathbf{v}^t_{i,j} / \sum w^t_{i,j}$ over all vertices in the wrinkle segment, where $w^t_{i,j}$ is the wrinkle probability associated with vertex $\mathbf{v}^t_{i,j}$.

4. *Back-buckle top* ($\mathbf{p}_2$): From human anatomy it is reasonable to assume the wrinkle folds inwards on an orbit around the eye. To compute $\mathbf{p}_2$ we thus rotate $\mathbf{p}_1$ inwards around the eye center by half the geodesic distance from $\mathbf{v}_{bottom}$ to the vertex closest to $\mathbf{p}_1$ computed in the rest pose. See Figure 3.11.c for a schematic depiction. The skin compresses due to micro-wrinkles. We account for this by adjusting the rotation magnitude by the area ratio between neighboring visible skin in the current frame and in the rest pose.

5. *Back-buckle bottom* ($\mathbf{p}_3$): Computed analogously to $\mathbf{p}_2$ by rotating $\mathbf{v}_{bottom}$ instead of $\mathbf{p}_1$.

Next we want to create the wrinkle as a membrane (Figure 3.11.d) that smoothly transitions into the visible part of the lid defined by the two vertices $\mathbf{v}_{top,bottom}$ and also closely approximates the three inner feature points ($\mathbf{p}_{1,2,3}$) defined above. To resolve the compression problem shown in Figure 3.9.b, vertices are allowed to move freely on the membrane surface, in order to relax and slide into the wrinkle. We achieve this by alternating between two stages. The first stage relaxes the vertices in the wrinkle area by applying one iteration of Laplacian smoothing, which optimizes the surface to reduce stretching. This moves vertices into the wrinkle, but also potentially pulls them away from the intended wrinkle shape. We therefore apply a second stage, where we find the nearest vertex on the membrane for every feature point and constrain their positions to the feature points, while again solving for the membrane energy. This second step pulls the surface back towards the desired shape. For both stages we use Neumann boundary conditions at the border of the wrinkle area to ensure a smooth transition into the visible part of the eyelid. We repeat the two stages six times after which we found the vertices to have relaxed inside the wrinkle.

Self-Intersection Handling. The aforementioned process is not guaranteed to be free of self-intersections. Specifically, the formed wrinkle might protrude out from the visible part of the eyelid, or the smoothing might cause the wrinkle to intersect with itself. Figure 3.12.a shows an extreme case for illustration. Our main concern is to prevent any visually distracting artifacts and thus we wish to resolve self-intersections that are visible from the outside. To efficiently test for and correct self-collisions we can leverage the

anatomy of the eyelid. The wrinkles form in such a way that skin farther away folds over skin closer to the eye. In terms of our eyelid model this means that vertices further down a cross-section should never occlude vertices which are higher up. Our algorithm sequentially traverses the vertices in the wrinkle area from $\mathbf{v}_{bottom}$ to $\mathbf{v}_{top}$, where the vertex indices decrease from *bottom* to *top* ($bottom > i > top$). For every vertex $\mathbf{v}_i$, the method tests if $\mathbf{v}_i$ is occluded by a lower part of the eyelid (i.e. a triangle that contains at least one vertex $\mathbf{v}_{k,j}$ with $k > i$). If the vertex is occluded, it is moved in front of the occluding triangle. Visibility is determined from the point of view of the main camera. Figure 3.12.b depicts two possible scenarios: while the method will not report a self-intersection for $\mathbf{v}_1$ since it is only occluded by higher up parts, it will correctly identify and correct $\mathbf{v}_2$, which is occluded by lower parts. Once all vertices of the eyelid have been processed, we reverse the order and evaluate the vertices relative to the center of the eye (Figure 3.12.c). The vertices are now traversed from $\mathbf{v}_{top}$ to $\mathbf{v}_{bottom}$ and the method checks if a vertex is occluded by parts higher up. We alternate between these two steps until no more occluded vertices detected, which is typically within 3 iterations. The resulting wrinkle is now guaranteed to be behind the visible surface (Figure 3.12.d).

3.3.3 Integration

Finally, we integrate the tracked eyelid with the full face, which is provided by Beeler et al. [2011]. Their method uses the concept of *anchor frames*, which states that during a facial performance certain expressions will re-occur and they thus propose to pick a reference frame that is similar enough to the anchor frames to be able to track directly to them. This concept is also very useful in our scenario, as we found that tracking from closed eyelids is preferable. We therefore pick a frame with a neutral expression and closed eyelids as reference frame and construct the eyelid mesh from this frame to facilitate the integration with the face mesh (Figure 3.5). The lid is naturally aligned to the face mesh and we can establish dense correspondences between the two. We then use the eyelid to drive the deformation of the face mesh in this area. Since we made sure that the boundary of our eyelid deforms in a compatible manner to the face mesh (Section 3.3.1) the integration is seamless.

As there are many eyelid wrinkles at the micro- and mesoscopic scales during deformation, we apply mesoscopic optimization and temporal smoothing following Beeler et al. [2011] to produce temporally consistent high frequency details seamlessly across the full face including the eyelids.

3.4 Results

Eyelids are extremely unique and can produce extremely different wrinkles. This variance is not only visible from person to person but the shape and temporal deformation of the eyelids also differ substantially between the left and right eye of the same person. To demonstrate this variance we captured both left and right eyelids of three subjects. We show a selection of wrinkle reconstructions in Figure 3.13, which includes both single and double wrinkles of varying intensity. The shape of the eyelid does not just differ due to wrinkling but also depends on the underlying eyeball, as can be seen in Figure 3.13.e, where the corneal bulge of the eye is visible on the eyelid, even though the eye is fully closed. We further demonstrate the variation of eyelids within the same person in Figure 3.15. Notice for both actors B and C that one eyelid has two wrinkles while the other has only one. Additionally we illustrate in Figure 3.15 that our eyelid reconstructions naturally complement high-resolution facial capture methods, as the eyelids fit seamlessly into the face, increasing the reconstruction fidelity.

In addition to the intricate shape details of static eyelids, eyelid wrinkles also exhibit strong variation in their temporal formation. Figure 3.14 shows how a wrinkle is formed over time. During wrinkling, skin is folded in a rolling manner, which can be best seen in the accompanying video.

The accurate location where wrinkles form is essential for faithful reproduction. Figure 3.16 shows an overlay of the eyelid onto the input image and demonstrates how well the formed wrinkles coincide with the captured data.

As a last example, we demonstrate how the captured eyelids may be used in the creation of a digital double for an actor. The result of our system is combined with the eyes of the actor provided by Bérard et al. [2014] and we manually complete the model by sculpting the interface between the eyelid and eye as well as adding eyelashes and eyebrows. The renders shown in Figure 3.17 were created using Renderman with built-in shaders.

Our experiments were run on a Windows *i*7 machine with 32GB RAM and input images of 1176×864, an eyelid template mesh of approximately 12,000 vertices and a face mesh of roughly 1 million vertices. For this setting, our average computation times per frame were 24.7 seconds per camera for the wrinkle probability map creation, 26.8 seconds per camera for flow correction, 5.2 seconds per eye for contour tracking, 11.3 seconds per eyelid for the deformation and 24.5 seconds per eyelid for the integration step.

3.5 Conclusion

We have presented the first method for detailed spatio-temporal reconstruction of eyelids. Our approach combines a geometric deformation model with image data, leveraging multi-view stereo, optical flow, contour tracking and wrinkle detection from local skin appearance. Our results demonstrate that the model is able to provide a high-resolution mesh that deforms over time, reflecting detailed dynamic skin features and plausible deformations even for regions that are occluded or undergo extreme deformations. As the eye region is essential for conveying emotions, we believe that our method is an important step towards capturing expressive facial performances and the creation of realistic digital doubles.

Limitations and Future Work. Currently, our pipeline is not fully automatic and relies on a few manual steps, such as initializing the contour tracker with a few hand-drawn contours, and specifying the principal direction of the wrinkles when creating the eyelid mesh. While these manual steps can be done in a few minutes and do not require artistic skills, we plan a fully automatic pipeline for the future. By design, we can only reconstruct wrinkles that are identified by the wrinkle probability map, which in turn depends on the underlying image quality. Low resolution, motion blur or low contrast can cause detection to fail and a more sophisticated means of computing and extracting the wrinkles would be required.

Some expressions such as extreme grinning or squinting can cause wrinkle formations that our method does not handle well. For example, wrinkles in the radial direction may be filtered out by our wrinkle detection scheme (Figure 3.18.a). Figure 3.18.b depicts a case in which the skin under the wrinkle is compressed and bulges outwards rather than inwards, contradicting the assumptions of our model. In the future, we would like to extend our model to handle such cases. The ability to separate wrinkles depends on resolution, both of the wrinkle map and the proxy geometry. Figure 3.18.c demonstrates how very close wrinkles may be incorrectly merged if insufficient resolution is used.

Furthermore, as we compute several data terms, such as the eyelid contours, relative to the front camera, we can only handle minor head rotations. While this is sufficient for many capture scenarios, such as helmet cameras, extending the method to allow for large head rotations could be an interesting avenue for future research. Finally, our system so far focuses on performance capture and replay. For future work, an interesting avenue would be to add animation control, use our data to automatically create convincing eyelid

rigs, and investigate performance and detail transfer of the eye region to virtual characters different than the actor, thereby bringing the expressiveness of virtual characters to a new level.

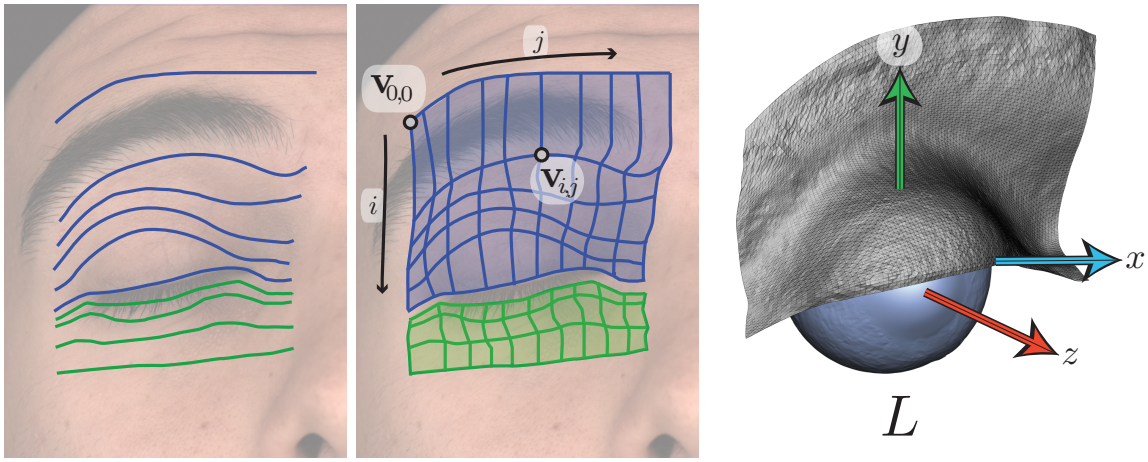


Figure 3.5: *The eyelid mesh is generated by manually drawing a few curves on the rest pose frame (left). From these contours a 2D grid is created (center) with the origin* $\mathbf{v}_{0,0}$ *in the top left corner, rows i running down and columns j to the right. From this grid the 3D eyelid mesh L is created using the depth maps and a reference coordinate frame is established (right).*

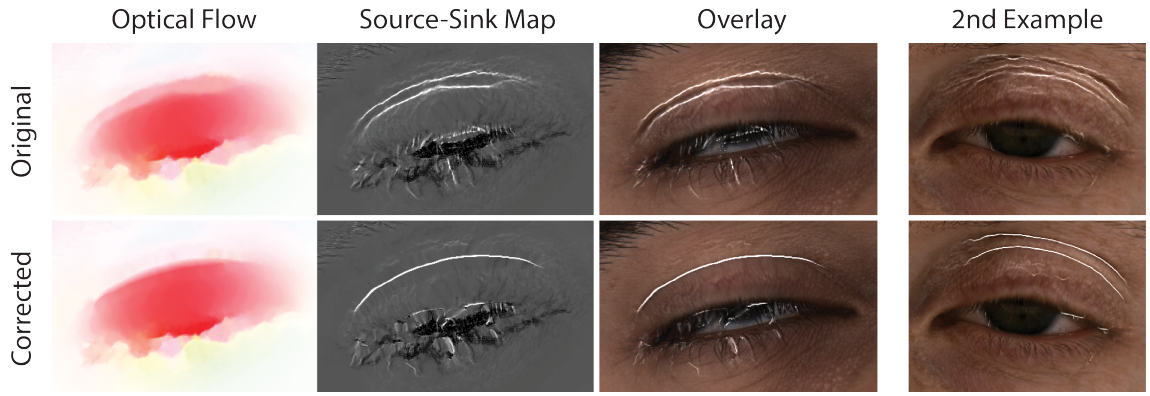


Figure 3.6: *Wrinkling poses problems for optical flow since the appearance changes and parts become occluded. The original flow shown in the top row is inaccurate around the wrinkle and compresses on both sides of the wrinkle, as shown in the source-sink map. Correcting the flow provides the desired behavior where the flow converges into the wrinkle instead. A second example with a double wrinkle is shown in the last column.*

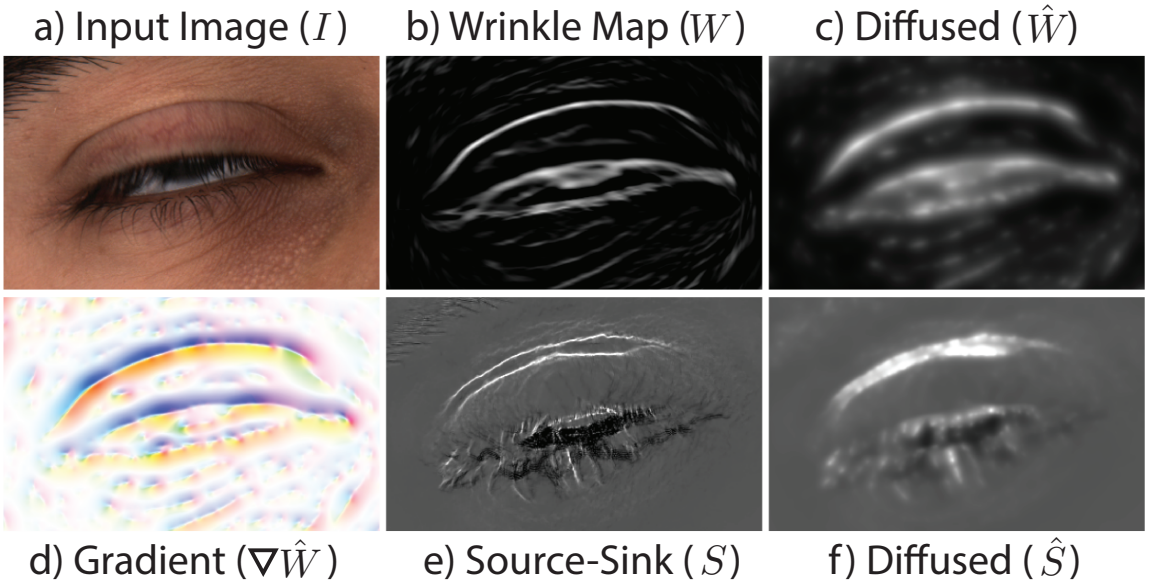


Figure 3.7: *To correct the flow maps we employ different variants of diffusion (Section 3.2.5). First the wrinkle probability map (b) is diffused isotropically with retention (c). The gradient of the diffused map (d) encodes the direction to the closest wrinkle and is employed to diffuse the source-sink map (e) anisotropically with retention (f) and finally the flow as shown in Figure 3.6.*

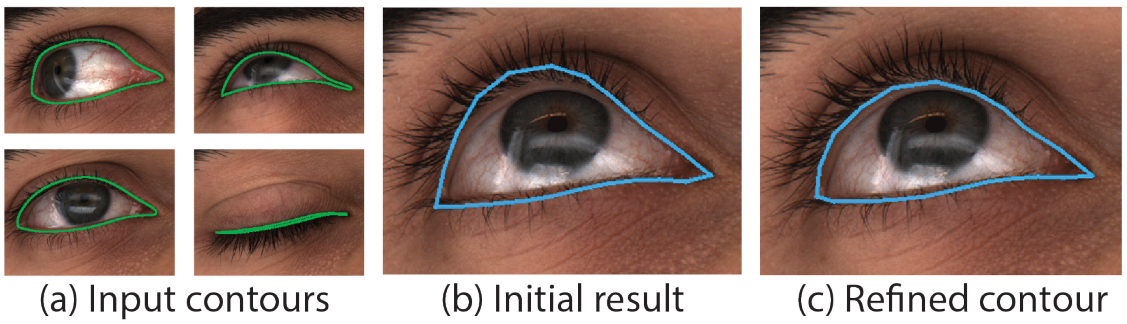


Figure 3.8: *Eyelid contour tracking pipeline: (a) subset of reference contours used for training, (b) initial tracking produces an estimate of the contour shape, but is not accurate enough for good localization, (c) the reference contour closest to the initial estimate is deformed using optical flow to refine the initial estimate.*

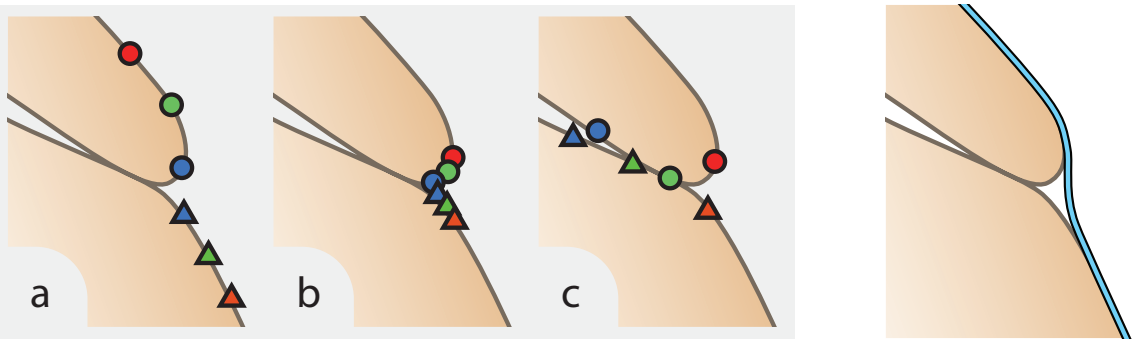


Figure 3.9: Left: *During wrinkling vertices are compressed from their initial position (a) into the wrinkle location (b) since they become occluded. Section 3.3.2 describes how the proposed deformation model moves them into the wrinkle in an anatomically plausible way (c).* Right: *Estimated depth (blue line) is inaccurate at the wrinkle location since the multi-view stereo method cannot resolve small scale details given the input image resolution.*

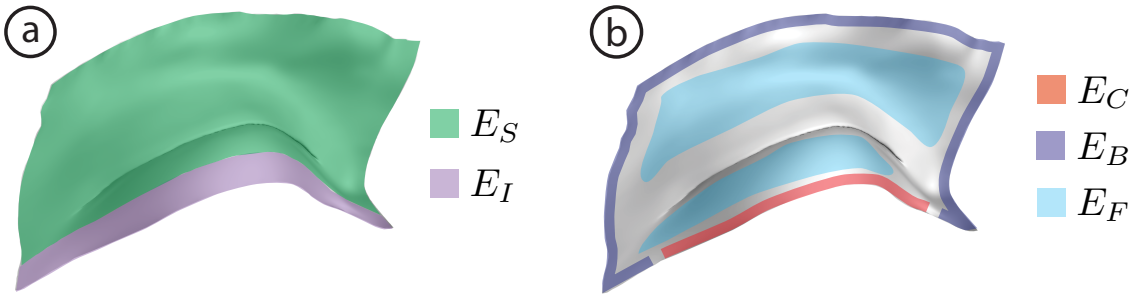


Figure 3.10: *(a) The visible skin deformation is regulated by two thin-shell energies: E_S regulates most of the eyelid for temporal smoothness, and E_I regulates the eye-eyelid interface for rigidity relative to the rest pose. (b) Regions contributing to each data term: tracked contours contribute to E_C, the interface with the face mesh to E_B, and visible interior regions contribute to the flow term E_F.*

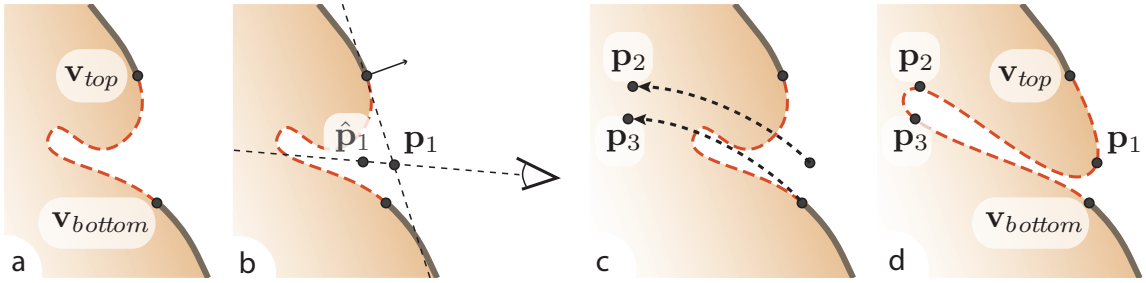


Figure 3.11: *(a) The top and bottom vertices bounding a wrinkle area in the cross section are defined as $\mathbf{v}_{top,bottom}$. These are the last vertices where we rely on the result from Section 3.3. (b) The front-buckle $\mathbf{p}_1$ is computed by projecting the weighted average $\hat{\mathbf{p}}_1$ of all visible vertices in the wrinkle area along the ray to the main camera onto the extension from $\mathbf{v}_{top}$. (c) The two back-buckle points $\mathbf{p}_{2,3}$ are computed by rotating $\mathbf{p}_1$ and $\mathbf{v}_{bottom}$ around the eye center. (d) The wrinkle is constructed as membrane from these feature points allowing the vertices in the wrinkle area to relax into the wrinkle.*

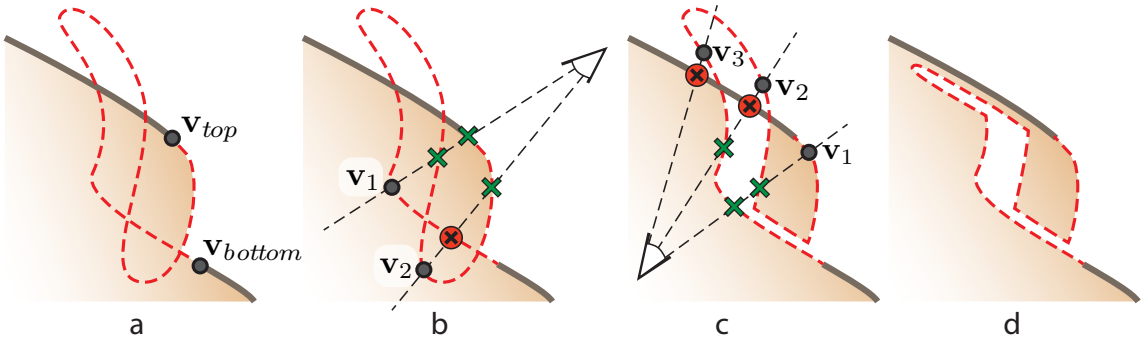


Figure 3.12: *The constructed wrinkle is not guaranteed to be free of self-intersections (a). The proposed method resolves self-intersections leveraging the fact that it is always the upper part of the lid folding over the lower. Thus we traverse the wrinkle area from bottom to top testing for occlusions by lower parts (b). We then move the occluded vertices in front of the occluding surface and reverse the procedure from top top bottom testing for occlusions with respect to the eye (c). Alternating these steps several times produces an intersection free wrinkle (d).*

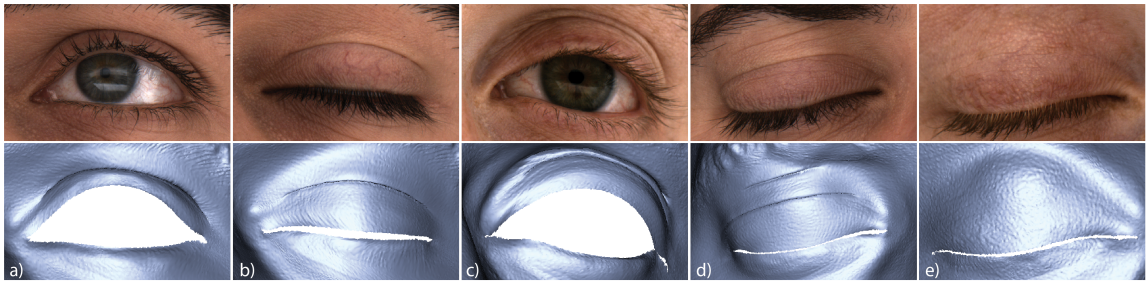


Figure 3.13: *We are able to reconstruct complex eyelids including a) thick wrinkles, b) thin wrinkles, c) double wrinkles close together, d) multiple distant wrinkles, and e) as an eye closes and wrinkles disappear completely, notice the subtle bulge on the lid caused by the cornea.*

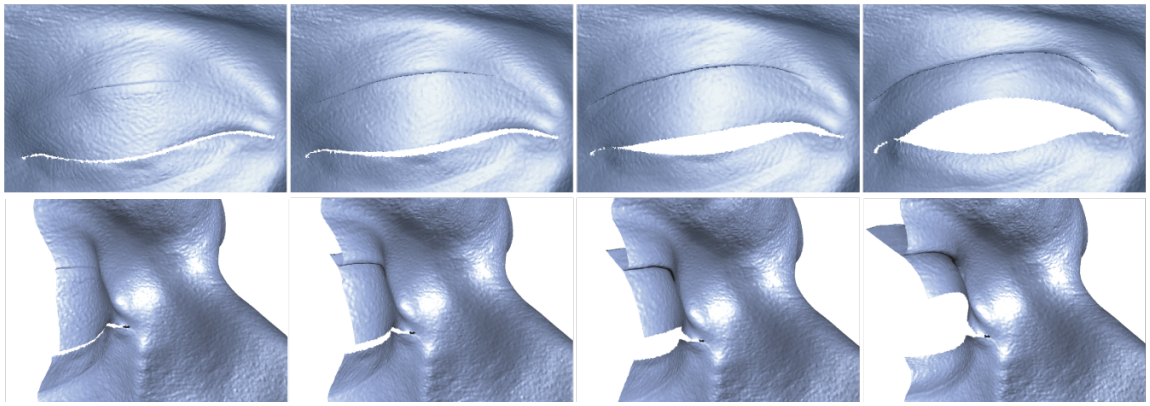


Figure 3.14: *Looking closely at the formation of a wrinkle we see the complex temporal dynamics of an eyelid. Our method is able to capture the skin folding under and creates a plausible eyelid shape, as seen from front (top) and from a side view cut-away (bottom).*

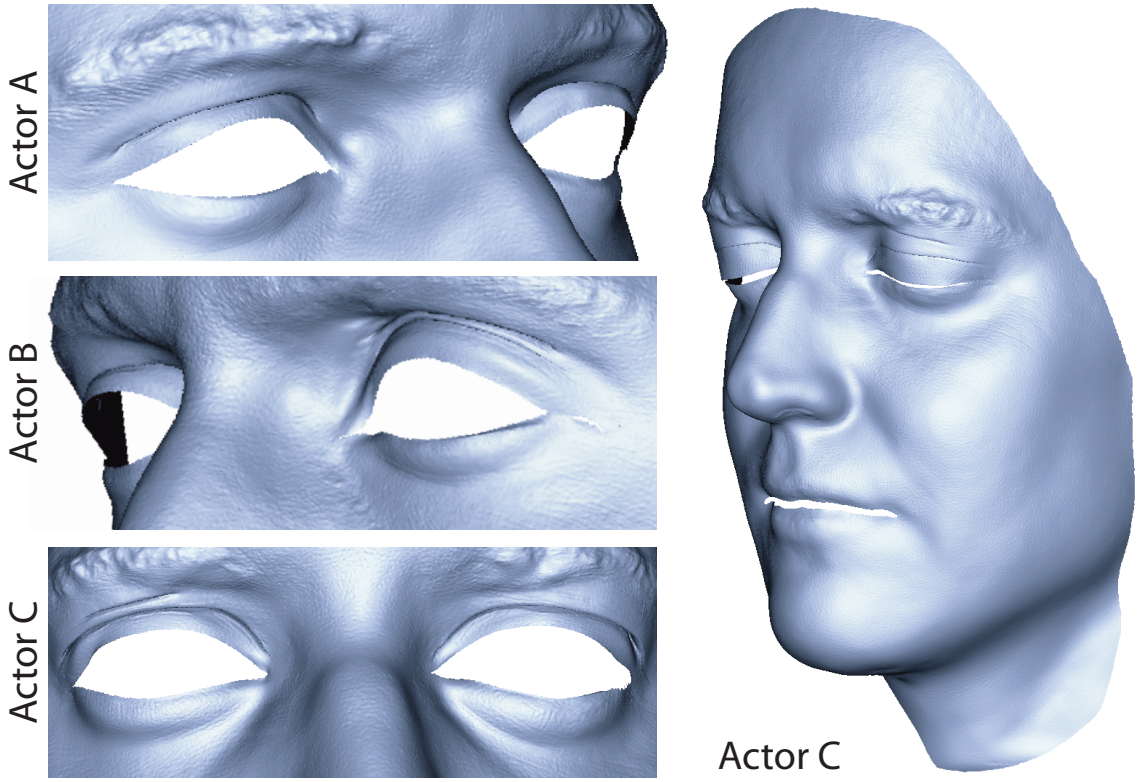


Figure 3.15: Left: *Eyelids can vary in shape quite substantially between people, as well as within the same person - notice the double wrinkles in one eye and single wrinkle in the other for both actor B and C.* Right: *Our reconstructed eyelids blend seamlessly with high-resolution captured faces.*

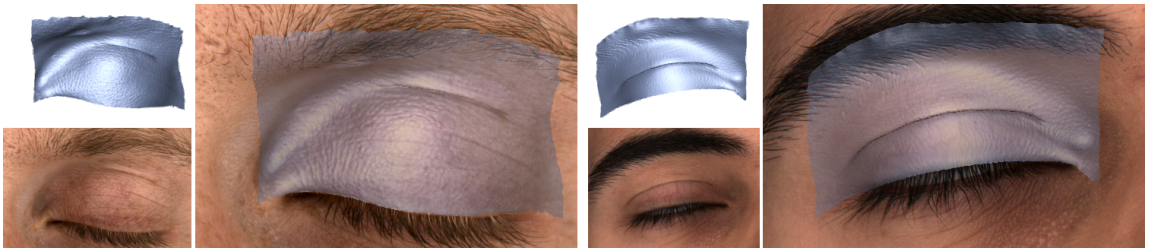


Figure 3.16: *We demonstrate the accuracy of our reconstructed eyelids by overlaying the mesh on an input image. The alignment of the wrinkles indicates the accuracy of the results.*

Figure 3.17: *Our eyelid reconstructions can be used to make high-fidelity digital doubles. Here we sculpted the thin interface to the eye, and added eyelashes, eyebrows, and eyes.*

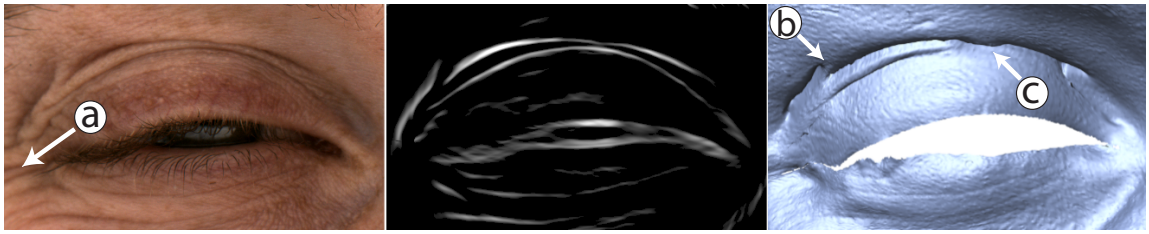


Figure 3.18: *A challenging grinning expression (left), the corresponding wrinkle map (center) and the reconstructed geometry (right). This expression pushes the limits of our method as radial wrinkles (a) are filtered out during the wrinkle map extraction, a skin crease is incorrectly modeled as an eyelid wrinkle (b) and very close wrinkles are merged during the geometry reconstruction (c).*

CHAPTER 4

Performance Enhancement

As explained in Chapter 1, realistic face modeling has long been considered a grand challenge in the field of computer graphics, for numerous reasons. Overcoming this challenge is also difficult since human faces can accommodate such a large range of expressiveness, from the most subtle hint of emotion to exaggerated exclamations. In real-life communication, subtle changes in facial deformation and dynamics can have a significant impact on the perceived expression and meaning conveyed by an individual. For example, a genuine smile may differ from a forced smile only in the slight tensing of one's cheeks. These subtle changes also contribute to the individuality that makes any particular person's face unique. Two different individuals may have a vastly different range and style of facial expressiveness.

A great deal of progress has been made toward solving this grand challenge, including sophisticated facial rigs, skin rendering algorithms, facial motion capture devices, and animation interfaces. However, despite these significant research contributions, creating synthetic facial performances that are as compelling and as expressive as a real actor's performance remains an elusive task. As the desired level of realism increases, animators must spend increasing amounts of time to incorporate the nuances of deformation that are characteristic of a particular actor's performance. At some point, the details become so subtle that they even elude the most skilled animators. Facial motion capture techniques based on marker-tracking, depth cameras like the Kinect, and fitting parametric models to video, also have a limited spatial and temporal resolution. Fine-scale details may escape the fidelity of the capture technology, especially when head-mounted devices are required.

These missing details contribute to an unfortunate result: many attempts at realistic facial animation fall prey to the "uncanny valley" effect (Section 1.1) and are perceived as eerie and lifeless.

In this Chapter, we contribute to the second step in the facial content creation pipeline - Augmentation (Section 1.2) - by targeting the subtle details of deformation and timing that escape both hand animation and motion capture systems and render an individual's facial performance unique and compelling. To this end, we propose a novel data-driven technique to enhance the expressiveness of facial geometry and motion. We start by recording a highly-detailed representative performance of an individual in which he or she explores the full range of facial expressiveness. From this data, our system extracts a model of expressiveness that encodes the subtleties of deformation specific to that individual. Once built, this model is used to automatically transfer these subtleties to lower-resolution facial animations that lack expressive details. The input animation will be augmented with the subtle deformations particular to the individual's face, increasing the perceived expressiveness and realism. Our system also takes advantage of the timing information in our database by enhancing facial keyframe interpolation so that the nonlinearities of expression formation exhibited by the real actor's performance are reflected in the interpolated and enhanced result. We demonstrate the robustness of our approach by enhancing a variety of input animations, including hand-animated facial rigs, face models driven by low-resolution motion capture data, morphable models animated using video data, and performance reconstructions generated with a Kinect using recent *faceshift* technology[1]. After our enhancement, the resulting animations exhibit the nuances and fine details of the original performance. Finally, we show that our algorithm outperforms the current state-of-the-art approach for data-driven facial performance synthesis [Ma+08].

In summary, the main contributions of this Chapters are:

- A framework for data-driven spatial enhancement of low-resolution facial animations, using a compressed shape space.
- A novel method for enhancing facial keyframe interpolation of temporal performances.
- Validation of our framework on four different types of input facial animations, with a direct comparison to state-of-the-art.

[1] www.faceshift.com

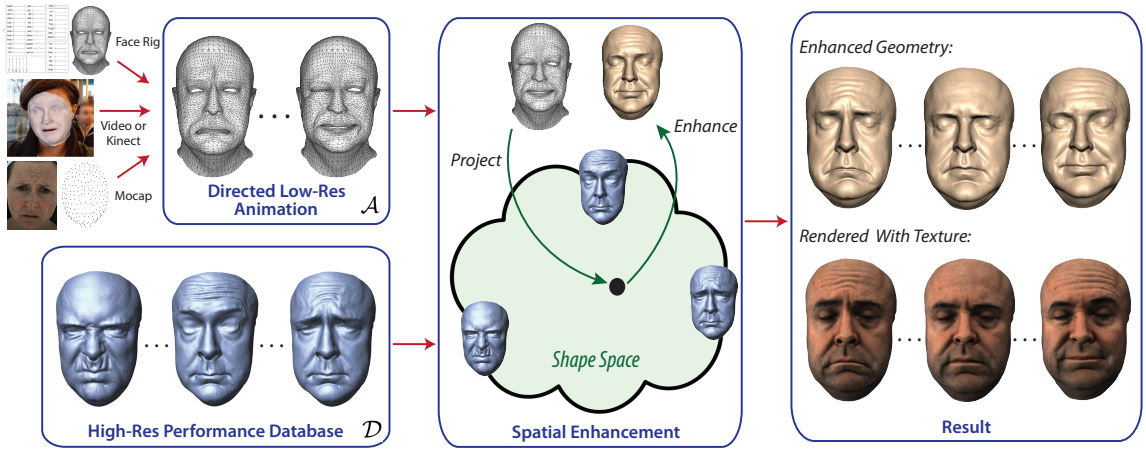


Figure 4.1: *Our spatial enhancement approach takes as input a low-resolution animation and a high-resolution performance database, and enhances the input animation with actor-specific facial details.*

4.1 Overview

Our goal is to enhance low-resolution facial performances by adding subtle facial features such as small wrinkles and pores, and/or temporal re-timing to match the dynamics of a real actor. The resulting animations should respect the underlying artistic content and enhance the expressiveness of the intended performance. This is achieved through the use of a high-resolution temporally coherent performance database, as illustrated in Figure 4.1.

Preprocessing. We intend to process low-resolution animation sequences that contain the creative intent of the animator or actor, but lack facial expressiveness and fine-scale details. In order to enhance the details, for a given actor, we build a dense performance capture database $\mathcal{D}$, consisting of $|\mathcal{D}| \approx 1000$ frames of facial geometry with consistent connectivity, captured using a high-resolution (e.g., pore-level detail) performance capture technique [Bee+11]. The database $\mathcal{D}$ is encoded into a *shape space*, which enables matching, projection and interpolation. This shape space is defined using the polar decompositions of the deformation gradients [SP04] with respect to a neutral frame $\mathbf{d}_0$, and thus effectively represents the stretching and rotation of each triangle (Section 4.2).

Spatial Performance Enhancement. Given an input animation $\mathcal{A}$, our enhancement algorithm (Section 4.3) combines its low-frequency components with the high-frequency components of corresponding frames from $\mathcal{D}$. For performance enhancement, each input frame is projected onto the shape space spanned by $\mathcal{D}$, and the relevant high-frequency components are in-

terpolated. These high-frequency details are then composed with their low-frequency counterparts, originating from the input frame, to generate the augmented mesh. The result is an upsampled version of the input animation, retaining the art-directed performance but enhanced with actor-specific expressiveness and details.

Temporal Performance Enhancement. Often, art-directed facial animations are created by hand, for example using a facial rig. In this case, it is common practice to represent the animation as a set of key-frames and then interpolate the in-between frames. Unfortunately, this interpolation may not match the true dynamics of the real actor, resulting in an unrealistic performance. As an added benefit of our performance enhancement system, we can augment the temporal component of the performance in a data-driven manner (Section 4.4). Keeping the artist in the loop when defining key-frames, we devise a new interpolation scheme to re-time the animation according to the actor-specific dynamics encoded in the database.

4.2 Preprocessing

Several preprocessing steps can be performed once per-actor. The database must be constructed (Section 4.2.1), and encoded into our shape space (Section 4.2.2), in a region-based manner (Section 4.2.3).

4.2.1 Performance Capture Database

We acquire a dense database of detailed facial geometry that includes pores, wrinkles and expressive deformations. The actor performs a number of short but expressive sequences that are stored in the database $\mathcal{D}$. The high-resolution geometry must be in full correspondence over time so that the motion and deformation of every point on the face is known. The database can be acquired using any high-resolution 3D facial performance capture method and we employ the passive approach of Beeler et al. [2011]. In this method, multi-view video sequences are recorded and high-resolution per-frame geometry is computed with the static reconstruction method of Beeler at al. [2010] at approximately 40 frames per second. We then temporally align multiple sequences using dense image-space matching and per-frame geometry propagation, yielding a temporally consistent database of 24 expressive performances (in full vertex correspondence). Figure 4.2 shows a subset of poses from the database for each of our two actors.

Figure 4.2: *Performance capture database samples for our two actors*

4.2.2 Data Encoding

Frequency Separation. Once the database is captured, we separate the low- and high-frequency components of $\mathcal{D}$ using a low-pass filter operation $f(\cdot)$. We create the dataset $f(\mathcal{D})$, where $f(\mathcal{M})$ for a set of meshes $\mathcal{M}$ denotes the set of all its filtered meshes, i.e., $f(\mathcal{M}) = \{f(\mathbf{m}) | \mathbf{m} \in \mathcal{M}\}$. In our work, this separation is conducted using implicit curvature flow [Des+99], an iterative approach where in each iteration we find new vertex positions X^{n+1} by solving the system:

$$
\begin{aligned}
(I - \lambda dtK)X^{n+1} &= X^n, \\
K_{ij} &= \begin{cases} -\frac{1}{4A_i}(cot\alpha_j + cot\beta_j) & i \neq j \\ -\sum_{k \neq i} K_{ik} & i = j \end{cases}
\end{aligned} \tag{4.1}
$$

where α_j and β_j are the two angles opposite to the edge in the two triangles having the edge e_{ij} in common, A_i is the sum of the areas of the triangles having x_i as a common vertex, and λdt was chosen to be 125 in all our experiments. By using a large λdt, we require only a single iteration to aggressively attenuate the high-frequency components, while preserving roughly the same levels for the low frequency part. During performance enhancement we will also separate the frequencies of the input animation (Section 4.3.1 and Figure 4.5), so this step creates a common ground for all different types of inputs presented in this work and enables an accurate matching and enhancement process. Note that this process preserves the size of input triangles, which might yield very small or nearly degenerated ones. To avoid the numerical instabilities caused by such triangles, we also perform one iteration of uniform Laplacian smoothing after the implicit fairing process.

Encoding. Finally, the database frames are encoded into the shape space. For every database frame $\mathbf{d} \in \mathcal{D}$, the deformation gradients encoding the difference between the mesh and its low-frequency counterpart $f(\mathbf{d}) \in f(\mathcal{D})$ are encoded and saved, denoted by $\mathbf{d}^h$. These high frequencies will be used as details, and are transferred onto the input animations during the spatial enhancement process (Section 4.3.4).

For the projection, we encode the low-frequency component relative to the low frequency neutral pose $f(\mathbf{d}_0)$. Since the input is fairly low resolution, the full shape space of the high-resolution mesh contains redundant information. To reduce the runtime and memory footprint, we encode into a *compressed* shape space. Rather than encoding the deformation gradient per-triangle, we uniformly cluster the high-resolution mesh into patches and encode only the average deformation gradient for each patch. Patches are computed using a random seed-and-grow approach. In practice, we found that patches of 100 vertices provided ample compression for our high resolution meshes. The shape space vector size is thus reduced by a factor of 100. Figure 4.3 illustrates the clustering on one of our datasets. We denote the set of all compressed shape space vectors in our database as $\tilde{\mathcal{D}}$. Note that alternative compression schemes are possible, for example mesh simplification. However, deformation gradients would have to be computed a second time on the simplified mesh, and therefore we use the clustering approach.

4.2.3 Regions

Tena et al. [2011] show that region-based face models generalize better than their holistic counterparts. Regions are a partition of the mesh faces into

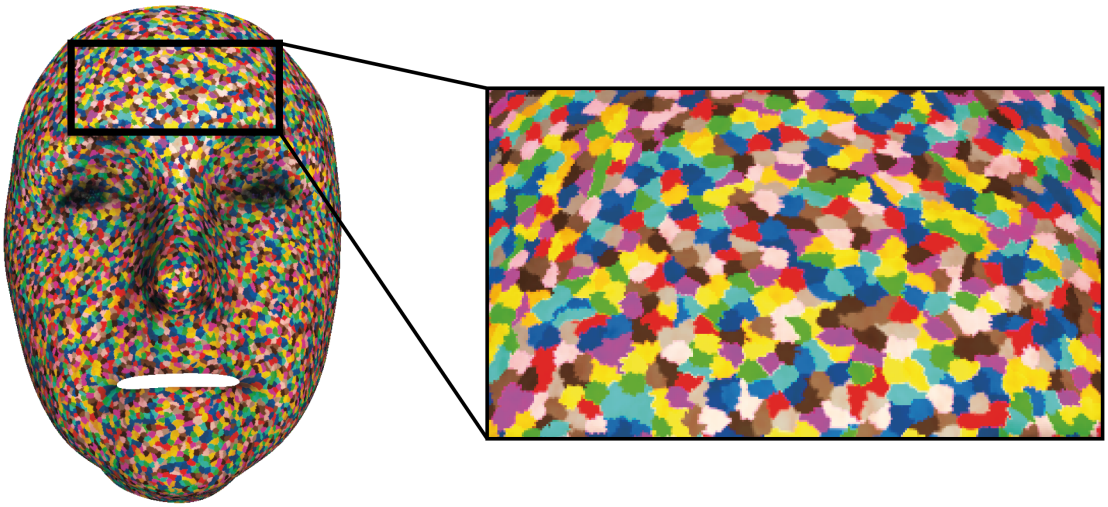


Figure 4.3: *Triangle clustering for encoding into a **compressed** shape space. We encode the average deformation gradient for each patch rather than per-triangle deformation gradients.*

what are usually groups with common functionality. The regions $\mathcal{R}$ we have chosen to use are based on Tena et al.'s work, which clusters the vertices according to their correlation in movement. While using the same clustering, we distinguish between *voluntary regions*, controlled by the actor directly, and *involuntary regions*. The latter are areas that deform indirectly, governed by the voluntary regions. As shown in Figure 4.4, out of the total 13 regions described by Tena and colleagues, we classify four as voluntary and use them in our matching process: left and right halves of the mouth, and the left and right eyebrows. This classification enables us to detect asymmetrical expressions as well as decoupling of the eyes and mouth. Note however, that as elaborated in Section 4.3.3 and decipted in Figure 4.4, this is a soft-boundary decoupling - each mesh triangle is weighted according to its geodesic distance from each of the regions. In Section 4.3.4, these regions will be used both for matching and blending of the high-frequency details originating from several database meshes.

4.3 Performance Enhancement Model

Our data-driven performance enhancement approach consists of five steps for each frame. First, the frame is brought into correspondence with the database geometry and we perform frequency separation (Section 4.3.1). Next, the frame is *encoded* into the shape space (Section 4.3.2). Then, we project it onto the shape space of the database in a *matching* step (Sec-

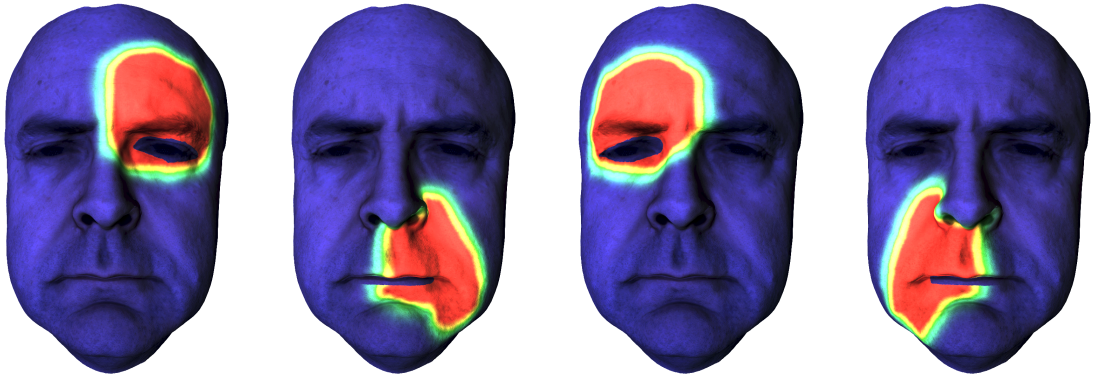


Figure 4.4: *The four voluntary regions that are used in the matching process. The weights smoothly decrease from* 1 *(red) to* 0 *(blue).*

tion 4.3.3). Based on the matching, we *interpolate* the relevant high-frequency details from the database and compose them with the low-frequency input mesh (Section 4.3.4). Finally, we *reconstruct* the resulting mesh, and assign per vertex colors to it by linearly interpolating the colors from the same database frames (Section 4.3.5).

4.3.1 Input Animation Pre-Processing

Input Sources. The input animations $\mathcal{A}$ can come from any source, but in industry these are most often created using a manually-controlled rig or driven by sparse marker-based motion capture. For one of our experiments, we use a facial rig as input built from a set of $B \approx 40$ blendshapes. These are based on static scans of an actor according to the facial action coding system (FACS) [EF78]. This rig can be fully controlled manually, allowing artists to create arbitrary animations. The rig only spans an approximate subset of facial expressions, and there is a natural limit on the accuracy and number of animation parameters an animator can evolve over time. Some example snapshots from an animation created using this facial rig are shown in Figure 4.10 (left column).

Additionally, we evaluate our algorithm on three other input sources, including sparse marker-based motion capture data, a low-resolution morphable model fit to a monocular video sequence [Dal+11], and a blendshape-based facial animation driven by Kinect data using *faceshift* [Wei+11].

Registration and Frequency Separation. The input animation $\mathcal{A}$ will have a different geometric structure than $\mathcal{D}$ (e.g. it could be a lower resolution mesh or possibly just marker positions). In our examples the actor is the same for $\mathcal{A}$ and $\mathcal{D}$, although this need not be the case. In theory, we could transfer

facial details to different actors, although in practice, the facial properties of the face in the database $\mathcal{D}$ and the animation $\mathcal{A}$ should be similar to achieve visually plausible results.

To be able to work in the same shape space and to provide a means for detail transfer, we obtain a dense correspondence between a database neutral frame $\mathbf{d}_0$ and the neutral frame $\mathbf{a}_0$ from our low resolution input source. We start by aligning $\mathbf{a}_0$ to $\mathbf{d}_0$ using the non-rigid registration method of Li et al. [2008]. This establishes a correspondence between the database meshes and the input data that we can propagate over the entire input. Naturally, the number of vertices n_d for a pose in $\mathcal{D}$ is much larger than the number of vertices n_a for a pose in $\mathcal{A}$. For example, in the case of the face rig, $n_a \approx 4000$, whereas $n_d \approx 1.2M$. We therefore employ a linear deformation model to propagate the animation specified by $\mathcal{A}$ to the registered high-resolution neutral pose [Bic+08b]. The resulting animation $\bar{\mathcal{A}}$ matches exactly the motion of $\mathcal{A}$ and is in dense correspondence with our database $\mathcal{D}$. We then separate the low frequencies in $\bar{\mathcal{A}}$ using the same procedure as with the database (Section 4.2.2), to obtain $f(\bar{\mathcal{A}})$. $f(\bar{\mathcal{A}})$ is now a standard form that is similar to $f(\mathcal{D})$, no matter the source of the original animation $\mathcal{A}$. The results of this process are decipted in Figure 4.5.

4.3.2 Encoding

As mentioned in Section 4.1, our performance enhancement uses deformation gradients [SP04]. In order to achieve accurate matching, every smoothed input frame $f(\tilde{\mathbf{a}}) \in f(\bar{\mathcal{A}})$ is first rigidly aligned to the database using the method of Horn [1987], and only then is the frame encoded. The deformation gradients are encoded with respect to a smoothed version of a database neutral pose, into a vector denoted $\mathbf{a}^{\ell}$. As explained in Section 4.2.2, a compressed version $\tilde{\mathbf{a}}$ of the same vector is created to be used in Section 4.3.3. The rotation $\mathbf{R}_a$ and translation $\mathbf{T}_a$ matrices that are produced during the alignment operation are also stored along with the deformation gradients vector, and all are used for reconstruction in Section 4.3.5.

4.3.3 Matching

In order to transfer the subtleties of facial details recorded in our database to a low-resolution input mesh we must locate the corresponding high-frequency data in our database. This can be formulated as a projection of the compressed shape space vector $\tilde{\mathbf{a}}$ of the input frame onto the database. In other words, we represent the input frame as a convex combination

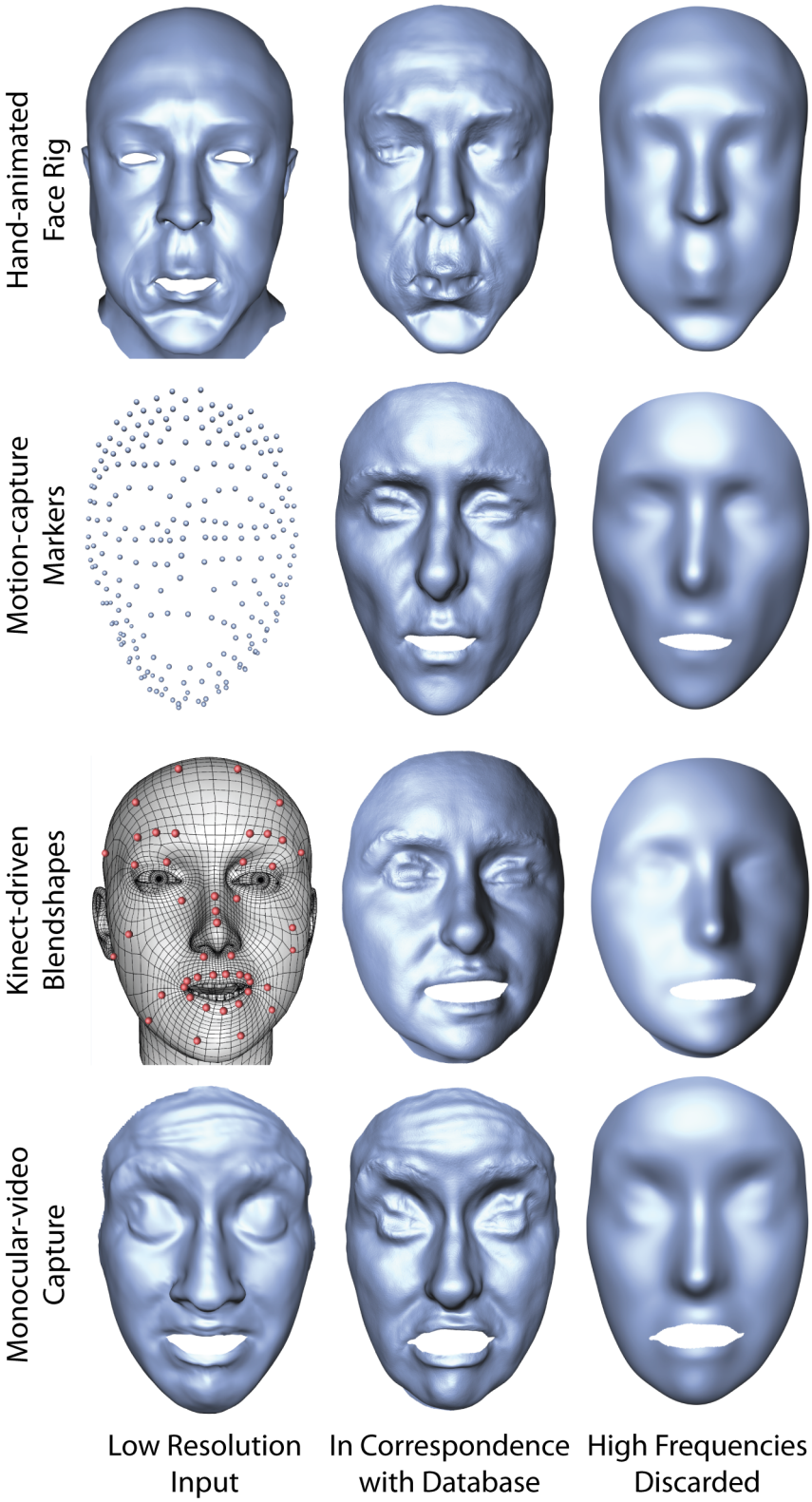


Figure 4.5: *Input frame pre-processing for all input sources (top to bottom): Hand animated rig, tracked mocap markers, depth camera driven rig and monocular video based capture. The input frame (left), drives a deformation of the database neutral pose (middle) and is standardized using smoothing (right).*

of weights $\mathbf{w}$ that represents a point in the shape space, spanned by the database, that is closest to the input frame, i.e.,

$$\min_{\mathbf{w}} \|(\tilde{\mathbf{D}} \cdot \mathbf{w}) - \tilde{\mathbf{a}})\|, s.t. \sum_{w_i \in \mathbf{w}} w_i = 1, w_i > 0 \tag{4.2}$$

where each column i in the matrix $\tilde{\mathbf{D}}$ represents the compressed shape space vector of frame i in our database.

Previous works have restricted this matching to affine weights [Bar+09]. However, in our experiments, affine weights distorted fine features such as pores and we therefore enforce convex weights, solving the resulting minimization problem using a QP solver. Note that the weights quickly fall off to nearly zero outside the immediate neighborhood of $\tilde{\mathbf{a}}$, yielding only a small number of relevant shapes to interpolate. Furthermore, in Section 4.2.3 we describe a partition of the face into *voluntary regions*, $\mathcal{R}$. This implies that throughout the matching process each region of the face is treated independently. However, actual facial expressions are not decoupled between regions, as a genuine smile is shown on the eyes as well as on the mouth. Therefore, in our method, each region is represented as a vector $\mathbf{!}_r$ of weights per mesh triangle. The triangle weights are constant within the region, and decay in a Gaussian way as the geodesic distance from this area grows. This means that, with diminishing influence, areas outside the region participate in the matching process, yielding a subtle coupling effect. We incorporate these weights in a weighted least squares manner, solving:

$$(\tilde{\mathbf{D}}^T \cdot diag(\mathbf{!}_r) \cdot \tilde{\mathbf{D}})\, \mathbf{w}^r = \tilde{\mathbf{D}}^T \cdot diag(\mathbf{!}_r)\, \tilde{\mathbf{a}} \tag{4.3}$$
$$\forall r \in \mathcal{R},$$

with the aforementioned convex constraints, where $diag(\mathbf{v})$ is the diagonal matrix with $\mathbf{v}$ on its diagonal. In order to save runtime and memory consumption, the matrices $(\tilde{\mathbf{D}}^T \cdot diag(\mathbf{!}_r) \cdot \tilde{\mathbf{D}})$ are precomputed per region, and only $\tilde{\mathbf{D}}^T \cdot diag(\mathbf{!}_r) \cdot \tilde{\mathbf{a}}$ are computed during the matching process.

4.3.4 Interpolation

Having computed the weights per region, we are now able to generate the highly detailed augmented mesh. The high-frequency details of the database are linearly blended according to the computed weights with respect to the region weights $\mathbf{!}_r$. Formally, given a triangle t and a component i of its deformation gradient, the interpolation scheme is:

$$\mathbf{b}^h = \sum_{r=0}^{|\mathcal{R}|} \mathbf{D}^h_{t_i}\, \mathbf{w}^r\, \frac{\mathbf{!}_{r,t}}{\sum_{\rho=0}^{|\mathcal{R}|} \mathbf{!}_{\rho,t}} \tag{4.4}$$

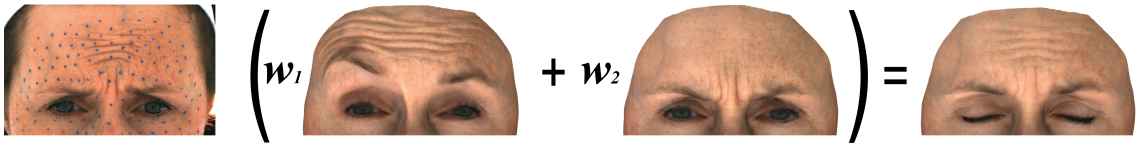


Figure 4.6: *To create the desired forehead wrinkles on the left, which are not present in the database, our enhancement technique blends between two different database frames.*

where $\mathbf{D}^h$ consists of columns which are the high-frequency detail vectors $\mathbf{d}^h$, $\mathbf{D}^h_{t_i}$ is the row in $\mathbf{D}^h$ that corresponds to the i-th component of the deformation gradient of triangle t, and $\mathbf{!}_{r,t}$ is the t-th element in the vector $\mathbf{!}_r$. Note that, per region r, the length of the matched weights vector $\mathbf{w}^r$ is $|\mathcal{D}|$ while the length of $\mathbf{!}_r$ is the number of triangles in a mesh. As mentioned above, this interpolation scheme blends the database within each region according to the matched weights and provides a normalized interpolation between regions. As a final step before reconstruction, the blended high-frequency details $\mathbf{b}^h$ are composed with the low-frequency deformation gradients of the input animation $\mathbf{a}^\ell$. This is done by converting the stretching and rotation vectors of $\mathbf{b}^h$ and $\mathbf{a}^\ell$ back to the deformation gradients' matrices and multiplying them. This process is equivalent to applying the blended deformation gradients representing the high-frequency details to the smoothed input animation mesh, as a deformation transfer [SP04], only without explicitly producing the intermediate low-frequency mesh.

4.3.5 Reconstruction

Having the final deformation gradients, we reconstruct the mesh using a slightly modified version of the Laplace-Beltrami operator [Bot+06]. First, in the interest of runtime performance, the Laplace-Beltrami operator is considered to be similar for all meshes, and so it is precomputed and prefactored once for the neutral pose. This assumption allows us to only use back-substitutions during reconstruction and has proven to be reasonable in all our experiments. Second, since the final deformation gradients are composed from several different ones, some artifacts tend to appear along the mesh boundaries. To suppress this artifact, we add a weighted regularization term to the reconstruction system of equations: in addition to the Laplace-Beltrami operator, we minimize the 1D Laplacian term along the mesh boundaries. In all our experiments, a minimal weight factor of $w = 0.05$ was sufficient to completely eliminate the artifacts.

After reconstructing the final mesh, we align it to the database using the method of Horn [1987] and we then apply the inverse transformations $\mathbf{R}_a^{-1}$ and $\mathbf{T}_a^{-1}$ that were calculated during encoding, to restore the mesh to its starting position.

As a final step, we perform the same interpolation scheme described in Section 4.3.4 on the vertex colors of the database meshes, and apply the result to the final mesh. An illustration of our enhancement technique is shown in Figure 4.6 for the forehead regions. Since the given expression is not in the database, two database frames are blended to create the closest match.

4.4 Temporal Performance Enhancement

In facial animation, the dynamic behavior of a performance greatly affects its perceived realism. Often, correct dynamics can be difficult to achieve. For example, when an animator creates a facial animation by rigging keyframes, the keyframes are interpolated linearly or with some hand adjusted ease-in/ease-out curves to create the full animation. This simple interpolation is insufficient to capture the timing of a real performance, and affects realism. In this section, we describe a method to automatically adjust the temporal behavior of the keyframe interpolation in a data-driven manner, using the previously described capture database.

The core concept that enables the temporal performance enhancement is the extension of the previously described frame projection to a sequence projection operator (Section 4.4.1). In short, given two sequences, this operator determines how one of the sequences can be approximated by the other, and how close the approximation is.

To start the process, the artist picks two keyframes they wish to interpolate, as well as the length of the desired motion. These two keyframes are then treated as a sequence of length two, and are projected onto our compressed shape space using the sequence projection operator to find the closest matching sequence. The temporal behavior of the matched sequence is analyzed, and this information is used to generate a well-timed interpolation of the input frames (Section 4.4.2). The result is an animation that closely follows the data-driven dynamic behavior while maintaining the user's artistic intent and spatial features. Note that the process is invoked by request of the artist, since temporal performance modification may not be desired in every scenario.

4.4.1 Sequence Projection

Sequence projection is an extension of the single frame projection operator described in Section 4.3.3. Given an input sequence, we wish to find the closest sequence in the database. This is a non-trivial task since the sequences in the database can have different dynamics and temporal behavior, resulting in different timing. For the sake of simplicity, we disregard the partitioning of the face into regions in this explanation, although the method is applied to each region independently.

As mentioned in Section 4.2.1, our capture database consists of sequences transitioning from the neutral pose to an extremity and back. We consider each such sequence as a temporal continuum, sampled uniformly at the sequence frames. The task of projecting an input sequence, α, onto such a sequence in the database, β, is simply one of finding a *valid* mapping between each frame of α to the time-line defined by β. A valid mapping is one that preserves the temporal order: a later frame in the input sequence α must be projected onto a later point in the time-line defined by the database sequence β.

Given an input sequence α, consisting of m frames $\{\alpha_0 ... \alpha_{m-1}\}$, and a database sequence β, consisting of n frames $\{\beta_0 ... \beta_{n-1}\}$, we start by projecting each frame α_i onto each of the linear segments $\{\beta_j, \beta_{j+1}\} \subset \beta$, and store the resulting blend weights as a matrix $\mathbf{T}$, as well as the distance (or error) of the projected points from the original ones as a matrix $\mathbf{E}$:

$$\begin{aligned} \mathbf{T}_{i,j} &= \text{Proj}(\alpha_i, \{\beta_j, \beta_{j+1}\})_0, \qquad 0 \leq i < m,\ 0 \leq j < n-1 \\ \mathbf{E}_{i,j} &= \| \alpha_i - \left(\mathbf{T}_{i,j}\beta_j + (1 - \mathbf{T}_{i,j})\beta_{j+1} \right) \| \end{aligned} \tag{4.5}$$

where $\text{Proj}(\mathbf{v}, \mathcal{S})$ is a vector of blend weights that represents the static projection (Section 4.3.3) of vector $\mathbf{v}$ on the set $\mathcal{S}$, and $\text{Proj}(\mathbf{v}, \mathcal{S})_0$ is the first element of this vector.

Next, we wish to identify the *valid* mapping P that yields the minimum error. This means that we search for a monotonic function that assigns a segment $\{\beta_j, \beta_{j+1}\}$ to every input frame α_i in a valid way, and minimizes the sum of projected distances. We solve for P that minimizes the following objective:

$$\begin{aligned} &\min_P \sum_{i=0}^{m-1} \mathbf{E}_{i,P(i)} \\ &s.t.\ P(i_1) \leq P(i_2) \qquad \forall i_1 < i_2. \end{aligned} \tag{4.6}$$

We solve this minimization problem using dynamic programming. This process is performed for all the sequences in the database, and the resulting projected sequence corresponding to the input is chosen to be the one yielding

the minimal error. Note that in most cases the input sequence α is projected only to a part of the chosen database sequence, which we refer to as the projected *sub-sequence*.

4.4.2 Temporally Driven Interpolation

We now describe how we use the sequence projection operator to interpolate the selected keyframes in a data-driven manner. As mentioned earlier, the user selects the desired length of the resulting sequence, m, and two endpoint keyframes of the resulting sequence, α_0 and α_{m-1}. As a first step, the sequence (of length two) $\{\alpha_0, \alpha_{m-1}\}$ is projected onto the database to find the closest sub-sequence. In case the input matches only a portion of a database sequence, the user may choose to *extend* the matched sub-sequence to the full corresponding database motion.

The matched sequence $\beta = \{\beta_0 ... \beta_{n-1}\}$ has exactly the dynamics we want, but typically contains a different number of frames n than the desired m. To rectify this, we start by creating a new sequence $\hat{\beta} = \{\hat{\beta}_0 ... \hat{\beta}_{m-1}\}$, which is a uniform interpolation of the ends of the selected subsequence $\{\beta_0, \beta_{n-1}\}$. Then, to get the right dynamics we wish to position each frame $\hat{\beta}_i$ on the continuous time-line uniformly sampled by β. This is accomplished by projecting $\hat{\beta}$ onto β, again using the sequence projection operator. If a frame $\hat{\beta}_i$ is mapped to the segment $\{\beta_j, \beta_{j+1}\}$ with blend weight t_i, one could deduce that $\hat{\beta}_i$ is projected to the point $j + (1 - t_i)$ in the continuous time-line. These time stamps are recorded for each frame.

As a final step, the two input keyframes α_0, α_{m-1} are linearly interpolated, creating the sequence $\hat{\alpha} = \{\hat{\alpha}_0 ... \hat{\alpha}_{m-1}\}$. The frames are assigned the previously computed time stamps $t(\hat{\beta}_i)$, forming a non-uniformly sampled piecewise linear time curve. The curve is then re-sampled using the shape space, in uniform intervals of $dt = n/m$ for the final animation. The result is a sequence composed solely of the artistically generated keyframes, but with the temporal behavior of the matched sequence of the database.

Figure 4.7 illustrates the result of the re-timing process, performed on a rigged transition between the rest pose and a smile. In this figure, we plot the time versus displacement of a vertex positioned on the edge of the mouth. An input linear interpolation is shown in red, and the closest matching database sequence is shown in dark blue. The database sequence contains the non-linear dynamics of the actor, but it represents a larger smile, i.e. the amount of displacement extends further than the input keyframe. The input sequence directly matches the first part of the database smile, where

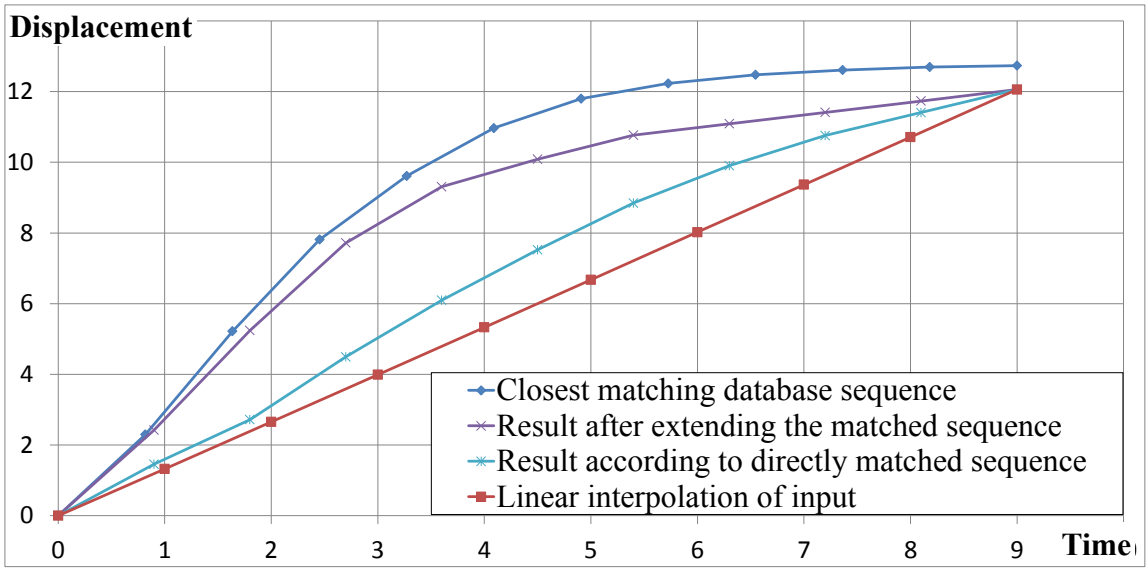


Figure 4.7: *A transition between the rest pose and a smile is enhanced using the temporal enhancement method. Here we show linear interpolation of one vertex (red), temporally augmented interpolation according to the closest matched sub-sequence (cyan), and temporally augmented interpolation according to the extended full sequence (purple). The actual temporal behavior of the database expression is presented for reference (blue).*

the motion is fast and relatively linear. If we re-time the input sequence using this direct match, we obtain the sub-sequence behavior shown in cyan, which is not much of an enhancement. However, should the artist choose to re-time the input after extending the matched sub-sequence to the full database smile, we obtain temporal behavior that matches the captured data as closely as possible, while maintaining the artistic intent (shown in purple).

The accompanied video exhibits a rigged transition of two key-frames, after being temporally and spatially enhanced. In order to keep the full integrity of the artistic intent, we propose to perform temporal enhancement before the spatial enhancement described in Section 4.3 is applied, and to enhance between only two interpolated keyframes at a time, although these are not constraints of the method.

4.5 Results

Our dynamic performance enhancement algorithm increases realism in facial animations by adding fine-scale details and expressiveness to low-

Figure 4.8: *Validation of our performance enhancement on a down-sampled high-resolution performance. Left: select frames from the high-resolution scan. Middle: downsampled inputs to the algorithm. Right: our enhanced result very closely matches the original.*

resolution performances. In order to validate our technique, we captured a high-resolution performance of an actor with the same reconstruction technique that we used to build the expression database. The performance is then spatially downsampled to mimic a typical input that we would expect, lacking expressive details. We then enhance the performance using our technique, yielding a result very similar in detail to the ground truth input scans. Figure 4.8 shows a few frames of the resulting validation. Minor differences between the ground truth and enhanced meshes (for example, around the mouth) are only visible where the correct shape is simply not in the database. For all examples in this section, we invite the reader to refer to the accompanying video for more results.

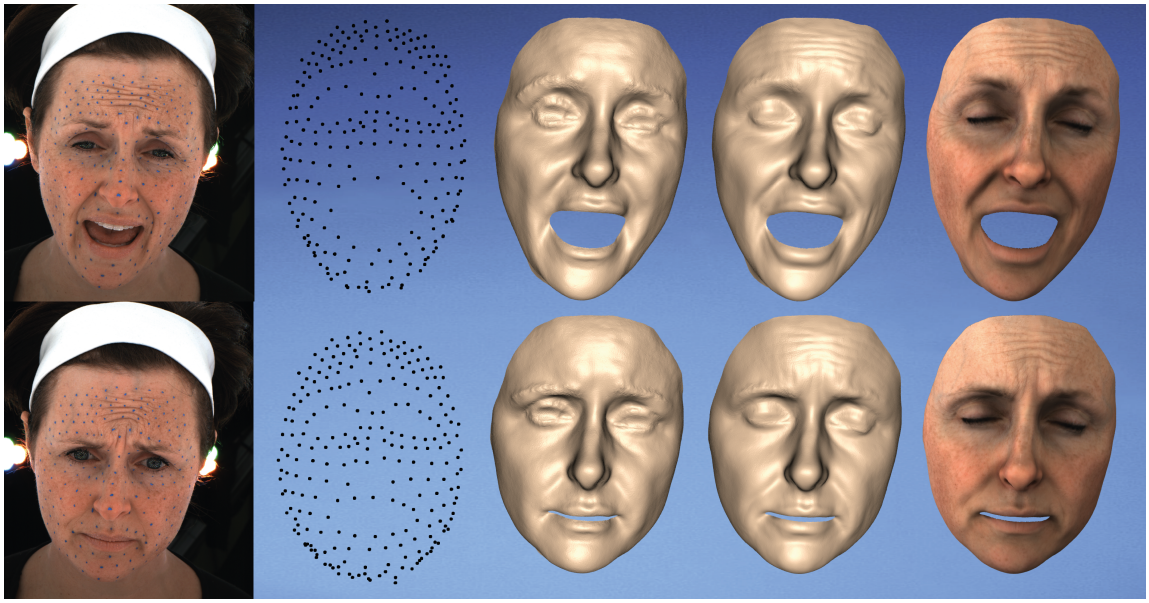


Figure 4.9: *Enhancing a marker-based motion captured performance. The columns from left to right: selected frames from the input sequence, tracked marker positions, traditional mocap result using the tracked markers to deform the mesh with a linear shell (notice the missing expression wrinkles), our enhanced geometry including expression details, final result rendered with texture.*

We demonstrate the robustness and flexibility of our enhancement algorithm by augmenting facial performances generated using four radically different facial animation techniques commonly used in industry and research. First, in Figure 4.9, we enhance a traditional marker-based motion capture animation. Approximately 250 markers are tracked and used to drive a low-resolution facial animation. The traditional motion capture approach is to deform a face mesh (e.g., using a linear shell model) with the marker positions as constraints. As a result, fine-scale details are clearly missing since they cannot be reconstructed from such a sparse set of markers (Figure 4.9, third column). Our technique is able to enhance the result with detailed wrinkles (Figure 4.9, fourth column), greatly adding to the expressiveness of the performance. We also illustrate the result rendered with per-frame reconstructed textures (Figure 4.9, last column). Note that we purposely do not target eye motion in the enhancement algorithm, and so we choose a single capture frame with closed eyes and blend the eye-regions into all final results using our interpolation framework.

Another common art-directable facial animation approach is a hand-animated rig. An example rigged performance and our enhanced result is shown in Figure 4.10. Most rig animations also lack fine-scale expres-

sion details, as it is time-consuming and difficult for animators to author these subtle effects. Our enhancement approach successfully adds the high-frequency details automatically. To illustrate the result of shape space matching, Figure 4.11 shows some of the closest database poses that are used in the region-based interpolation for one of the rig result frames.

A third mode of facial animation that lends itself to our enhancement technique is monocular face tracking using a morphable model [Bla+03; Vla+05; Dal+11]. Here, a low-resolution face model is automatically fitted to a video stream, which can be captured from a handheld camera in outdoor and remote environments (see Figure 4.12, left). By upsampling this type of animation (Figure 4.12, right), we demonstrate the ability to achieve studio-quality facial performance capture, even on a moving train. We believe this technology is a large step towards on-set markerless facial motion capture, which can benefit the visual effects industry.

Finally, we show that our algorithm can enhance facial animations captured using a Kinect depth sensor. Our input is generated from recent technology designed by *faceshift*, based on real-time performance-based facial animation [Wei+11]. An actor's facial motions drive a low-resolution blendshape model, which we then sparsely sample at 40 locations (see Figure 4.5) and enhance with our technique. Since the blendshape model is only an approximation of the performance, this result demonstrates the robustness of our approach to handle inaccurately tracked face motion.

The processing time of our algorithm is approximately 30 seconds per frame for our female actress with a mesh resolution of 500K vertices and 55 seconds for the male actor with a mesh resolution of 850K vertices, measured on an i7 desktop machine with 12GB of memory. We use the MOSEK [AA00] library to solve the QP problem (Equation 4.2), and process all four face regions in parallel. Realistic face renders are created using DAZ Studio with the *Elite Human Surface Shader*[2].

Comparison to Ma et al. [2008] Our work is most similar to the polynomial displacement map (PDM) technique of Ma et al. [2008], however our method contains some important benefits. The PDM technique is designed for real-time performance on well-tracked input sequences that lie inside the convex hull of a small training set. In that situation, our respective algorithms will produce similar upsampled results. However, in the case that the input shapes are far away from the training set, polynomial extrapolation artifacts can be seen in the method of Ma et al. (see Figure 4.14). Here we show our algorithm compared to an implementation of the PDM method with the

[2] www.daz3d.com

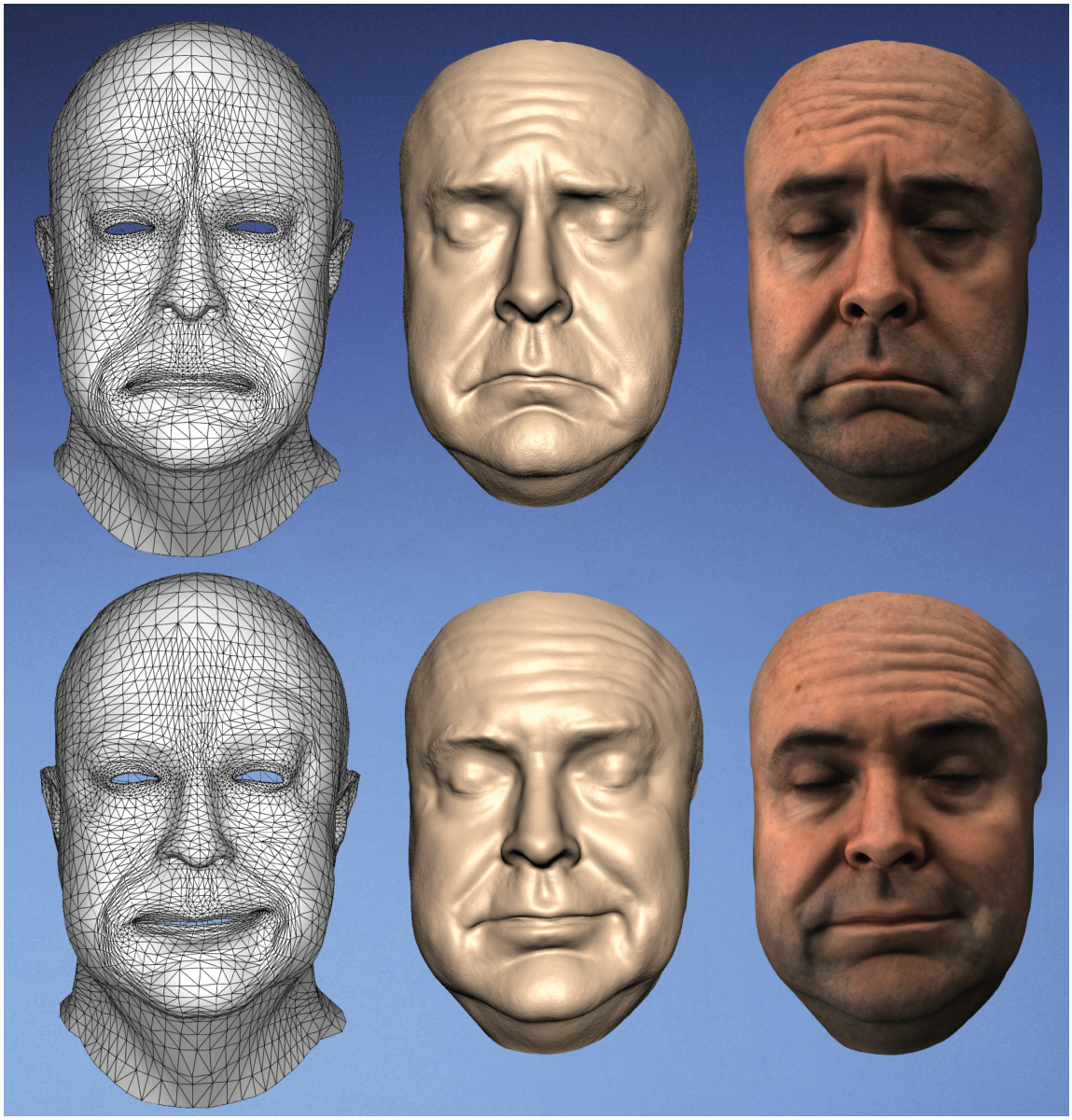


Figure 4.10: *Our method can enhance a rigged facial performance (left), adding the subtle details of expression particular to an individual's face (shown as a surface and textured).*

Figure 4.11: *Illustrating the shape space matching for one frame of the face rig result from Figure 4.10. Here we see four of the database poses that are used for interpolation.*

same database on two different inputs, one from Kinect input using faceshift and the other from motion-capture markers. The PDM approach produces unrealistic deformation of the face and amplification of the pore details. One could argue that increasing the size of the training set is a solution, however the PDM's are determined by an underlying vector field and discontinuities in the vector field causes artifacts in the resulting displacements. The bigger the training set, the harder this vector field is to control. This effect accounts for the discontinuities in the left part of the lip, the left cheek and the forehead. Finally, in the case of lower-accuracy input sequences like the ones from faceshift (top row of Figure 4.14), the input deviates again from the training set and results in more artifacts with the PDM approach. Our technique is more general and handles a wider range of scenarios.

Limitations and Future Work. One area for future work is to analyze and correct low-frequency errors. Currently, we assume that the low-frequency component of the input animation is correct, however it could be the case that it does not match the shape and dynamics of the real actor. Furthermore,

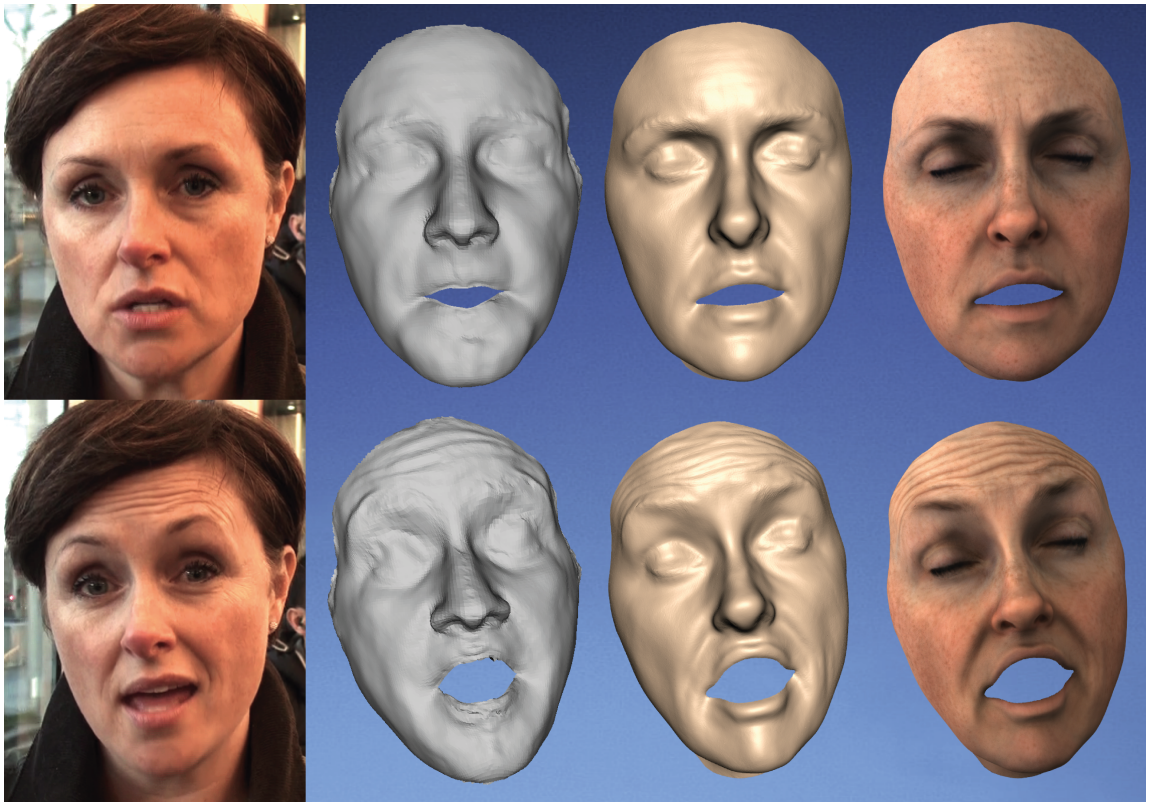


Figure 4.12: *Result on a morphable model fit to a monocular video sequence [Dale et al. 2011]. From left to right: selected frames from the video, the low-resolution fit model in gray, our enhanced geometry, and our final result rendered with texture.*

in this work we use the same actor for the database and input animations to ensure the facial properties are similar. An interesting avenue would be to explore performance transfer, by using input animations from one actor with a database from another. In addition, our method does not currently handle the eye region correctly due to a lack of accurate data in this area. This could be corrected with an improved acquisition system for the database. As a result, we blend closed-eyes into all results, which is easily accomplished with our shape space interpolation framework.

4.6 Conclusions

We have targeted the gap between low-resolution artistically created facial animations and high-resolution expressive performance capture. On the one hand, art-directed animations are attractive because the animation can easily be tuned for the desired performance, however they lack the subtle details of

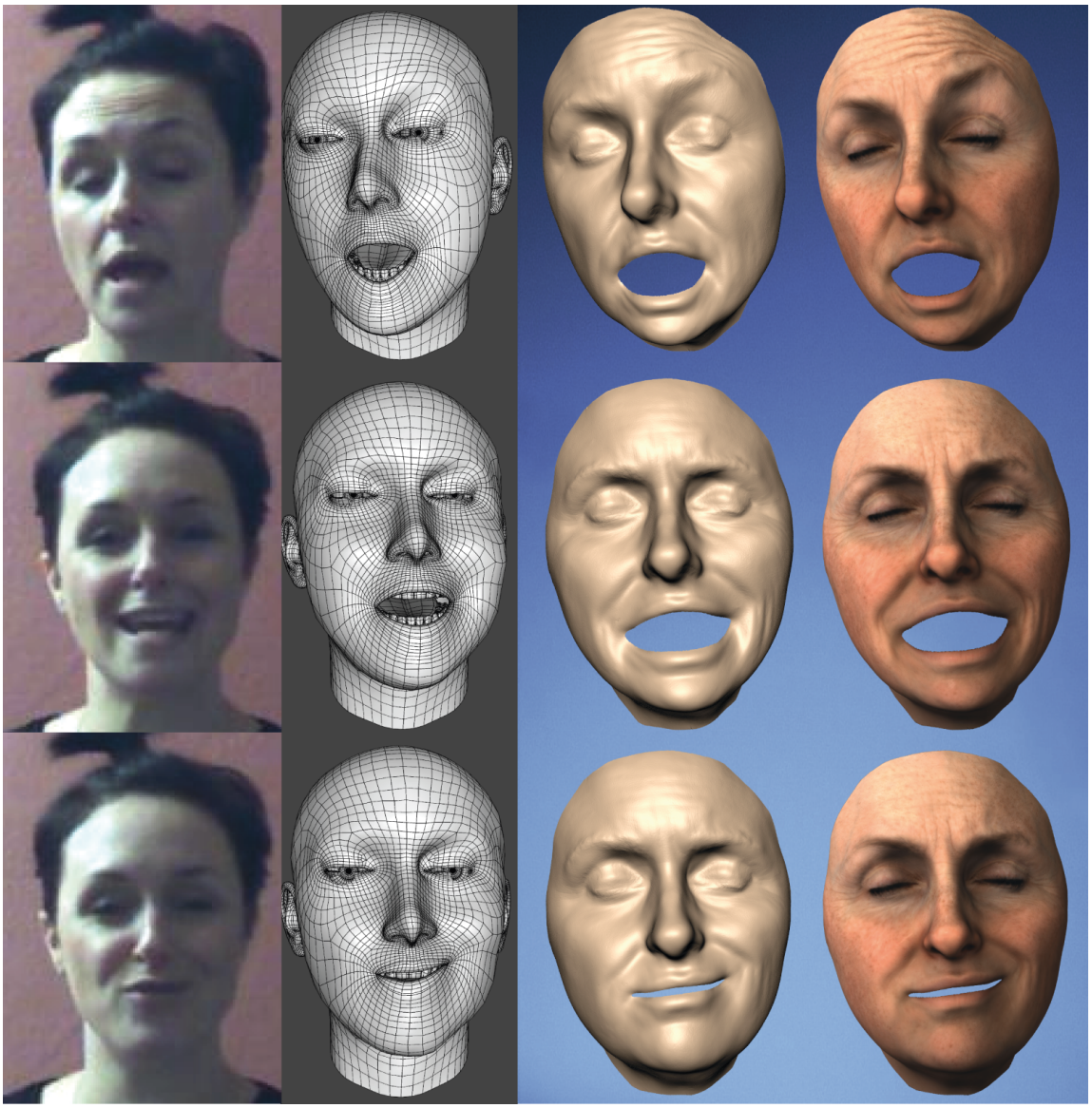


Figure 4.13: *Enhancement result on Kinect-driven input animations produced by faceshift [Wei+11]. From left to right: reference image from the Kinect, blendshape result of the faceshift software, our enhanced geometry, and our final result rendered with texture.*

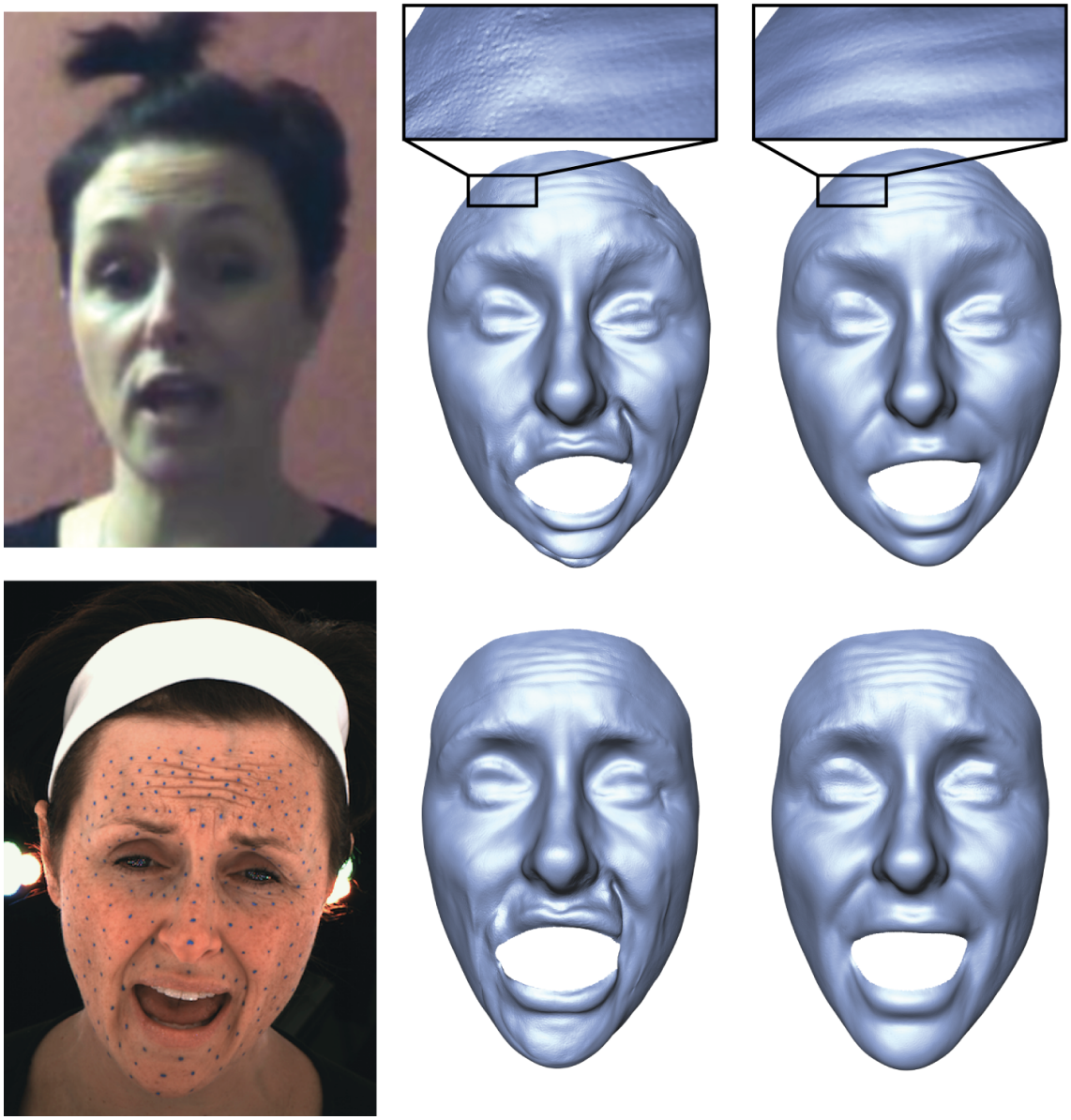


Figure 4.14: *Comparison to the Polynomial Displacement Map (PDM) technique [Ma+08] on two different datasets: Kinect input (top row) and motion capture markers (bottom row). The PDM method (center) exhibits more artifacts around the lips, cheek, forehead and exaggeration of pores, compared to our method (right).*

deformation and timing that make a real individual's facial performance so expressive and compelling. On the other hand, high-resolution performance capture can acquire the expressive facial details of a performance, but the result can only be played back without further directability. Our method extracts the fine-scale details from a performance capture database that spans the range of expressiveness for a particular individual, and then transfers these details to low-resolution input animations. Our system can also improve facial keyframe interpolation so that the dynamics of the real actor are reflected in the enhanced result. We demonstrate our method on four animations created by typical facial animation systems: marker-based motion capture, a hand-animated facial rig, a morphable model fit to monocular video, and a sequence reconstructed with a Kinect depth sensor. We also validate our result against ground truth data by using a smoothed performance capture animation as input, and provide a direct comparison to current state-of-the-art. With our technique, art-directed animations can now be enhanced to match the expressive quality of performance capture.

CHAPTER 5

Physical Avatars Augmentation

Bringing virtual characters to life is another one of the great challenges in computer graphics. While there were tremendous advancements in capturing, animating, and rendering realistic human faces in the past decade, displaying them on traditional screens conveys only a limited sense of physical presence. Animatronic figures or robotic avatars can bridge this gap. However, in contrast to virtual face models, reproducing detailed facial motions on an animatronic head is highly challenging due to physical constraints. Although steady progress in creating highly sophisticated robotic heads that strive to recreate convincing facial motions can be observed, for example those in Disney World's Hall of Presidents or "Geminoids" [NIH07a], these achieve only limited expressiveness when compared to a real human being.

Our goal in this Chapter is to significantly increase the expressiveness of such figures, and to allow to animate them and controlling their motion and appearance easily, by adding additional degrees of freedom with projected shading, thus improving the last step of the facial content creation pipeline - display (Section 1.2). An animatronic head consists of a deformable skin attached to an underlying rigid articulated structure. The appearance is determined by the material of the skin and its static texture. The articulated structure is driven by a set of motors, and their motion range determines the expressiveness of the figure. While adding additional mechanical components to extend the degrees of freedom would be an obvious choice, in practice this is often prohibitive due to the lack of space inside the head and the extensive cost. Instead, we suggest projected shading to obtain dynamic control of the appearance, and emulate expressive motion and appearance

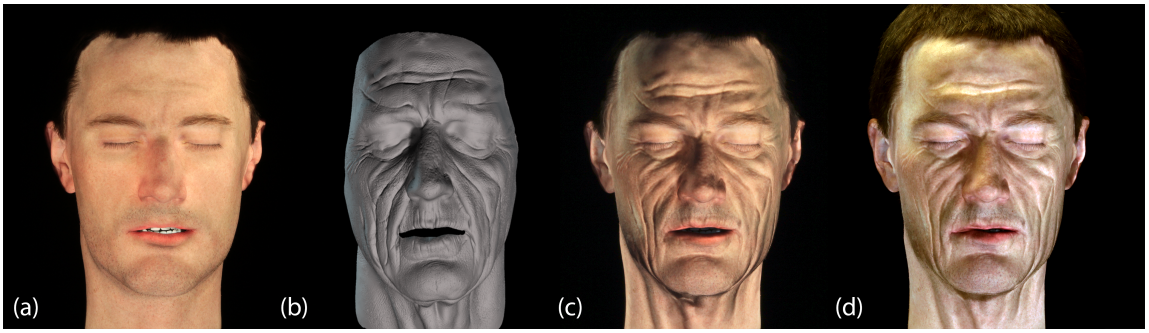


Figure 5.1: *Our system allows augmentation of a physical avatar (a) with projector-based illumination, significantly increasing its expressiveness. In (b) the target performance is shown. The appearance under controlled and ambient illumination is shown in (c) and (d).*

using a combination of low-frequency motion of the animatronic head and high-frequency shading.

In this Chapter, we present a two-scale model for representing facial motion tailored to animatronic heads, embedded in a multi-projection system. Low-frequency motions that can be reproduced by the physical head are represented as control parameters of actuators. High-frequency details and subtle motions that cannot be reproduced are emulated in texture space. In practice, we face the challenge that the mechanical motion range of the robotic head is significantly smaller than that of a human. However, the formation of facial details is strongly correlated to the underlying low-frequency motion. Given an arbitrary performance capture sequence, a naive baking of dynamic facial details into texture space would violate this correlation, due to the limited mechanical motion range. The robotic head would stop moving when reaching its limit, while the original input data would still contain motion and induce formation of facial details. We observed that this leads to visual artifacts. We therefore propose an efficient spatio-temporal method for decomposing the motion in gradient space, ensuring that we can reproduce the visual appearance of the sequence as close as possible while maintaining the correlation of low-frequency physical motion and formation of facial details. Using a multi-projector system, we then are able to convincingly and accurately replay the input animation.

More specifically, we start by acquiring a dense performance capture sequence of a person. First, we determine initial control parameters of the animatronic head that most closely resembles the target expression and acquire its detailed geometry for each frame. We then establish dense correspondence between the target performance and the performance of the animatronic head. Subsequently, we decompose the input performance into

low-frequency animatronic head motion and dynamic high-frequency shading details. Given the dense correspondence, we perform a space-time optimization that maps the input performance to the constraint motion gamut of the robotic head. Subsequently, we embed the high-frequency shading information on the robotic head geometry such that the low-frequency details conform in both performances.

Furthermore, we present a complete multi-camera and -projector system, allowing efficient optimization of the projection quality in terms of focus and contrast. Defocused projections and subsurface scattering lower the possibilities to reproduce high-frequency shading on the animatronic head. To maximize the overall contrast and focus, we present a model-based multi-projector optimization step to improve the final image quality considering physical light drop-off, smooth blending in overlapping regions, projection defocus, and subsurface-scattering. The optimization is carried out by carefully analyzing and modeling the required defocus and subsurface scattering properties independently of the actual pose of the animatronic. This has the advantage that it, in contrast to camera-based approaches, is independent of a particular viewing position and can be easily adapted to arbitrary animatronic poses without exhaustive per-frame data acquisition.

We implemented a prototype and demonstrate several results with our system. In all our results one can observe that our approach significantly increases the expressiveness of the animatronic head. We also show how our system can be used for artistic effects such as aging of faces, an application that would not be possible without projector-based shading.

5.1 Overview

Our approach on augmenting physical avatars using projector-based illumination starts by acquiring a source performance. For each input frame independently, we optimize for the animatronic head's actuation parameters that best resemble the input motion in simulation. Our goal is then to register the animatronic head to our projection-camera system, acquire information about its deformation behavior and subsurface-scattering as well as projector defocus to model the multi-projector light transport, perform a spatio-temporal decomposition and optimization of the head's motion and its texture to reproduce the desired facial performance, and finally, to reproduce the performance based on synchronized motion of the physical head and projection. An overview of the processing pipeline is given in Figure 5.2.

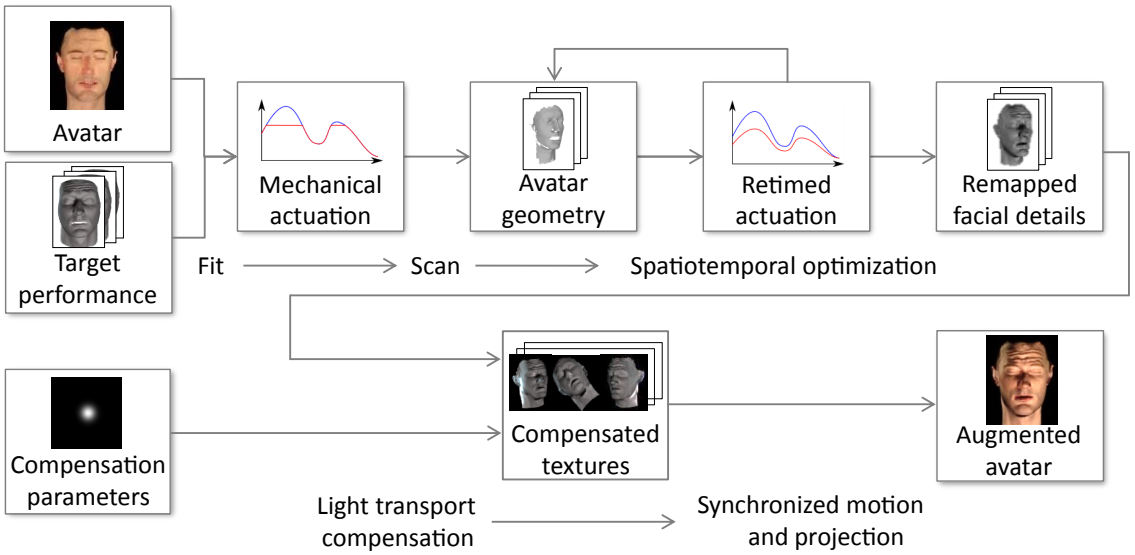


Figure 5.2: *Overview of the processing pipeline. A target performance drives the animatronics actuation, which is scanned by the system. Based on this data, the actuation parameters of the head are remapped to match the dynamics of the target performance. Next, the target performance is remapped onto the re-timed performance and its high-frequency details are embedded as colors. The sequence is then rendered from the calibrated projectors' point of view and globally optimized to compensate for light drop-off, defocus, and subsurface scattering. Finally the resulting images are projected onto the animated animatronic head.*

5.2 Performance Remapping

As input to our system we use a facial animation sequence that was captured using the system of Beeler et al. [2011]. It provides a detailed mesh sequence with explicit temporal correspondence. Our avatar is a proprietary animatronic head developed by Walt Disney Imagineering. It is driven by electric motors and features 13 parameters to control the actuation of the skin. We treat the underlying mechanical structure as a black box and use a finite-element-based optimization approach to determine the parameters for matching the deformation of the skin to each frame of the input sequence in a least-squares sense as described in Bickel et al. [2012]. We then place the animatronic head in our projector-camera system and acquire and register its dense performance as described in the following subsection. Furthermore, we then re-time its performance, as described in Section 5.2.3.

5.2.1 Geometry Acquisition

Accurate projection and remapping requires an accurate 3D representation of the physical avatar.

Acquisition setup. In order to enable the system to be self-contained once it is deployed, we used five calibrated cameras to capture structured light patterns for projector calibration, 3D reconstruction, and defocus estimation. The complete setup is depicted in Figure 5.3. The cameras are geometrically calibrated using a standard checkerboard-based calibration technique [Zha00]. A series of structured light patterns, consisting of gray codes and binary blobs, is used to get a sub-pixel accurate mapping from camera to projector pixels. We then generate a medium-resolution 3D point cloud $\mathcal{P}_n$ for each frame $n = 1...N$ of the animatronic head's performance as described in [HZ04]. Using direct linear transformation with non-linear optimization and distortion estimation enables an accurate calibration of the projectors. While the data provided by the scans is relatively accurate and represents the motion of the animatronic head well, it is incomplete in terms of both density and coverage: Regions that are not visible to more than one camera (due to occlusion or field of view) are not acquired at all, or yield a sparse and less accurate distribution of samples. Instead of adding more cameras to the system, we opted to scan the neutral pose once before deployment with a high-quality scanner [Bee+11], and then to complete the missing data using non-rigid registration.

Non-rigid registration. Given the acquired point-clouds $\mathcal{P}_n$, we generate a complete detailed mesh sequence $\mathcal{M}_n$, using the high-quality scan of the neutral pose (denoted by $\mathcal{N}$). We achieve this by deforming $\mathcal{N}$ to match the point-cloud $\mathcal{P}_n$ in all high-confidence regions. For this, we first convert the point-cloud $\mathcal{P}_n$ to a manifold mesh $\hat{\mathcal{P}}_n$, by employing Poisson reconstruction [KBH06]. Using a similarity matching criterion combining distance, curvature, and surface normal as recommended in Tena et al. [2006], we then automatically find correspondences between $\hat{\mathcal{P}}_n$ and $\mathcal{N}$. The aforementioned process yields semantically plausible correspondences only for relatively small variations between meshes. Therefore, we use an incremental tracking process. For each frame n with corresponding acquired point-cloud $\mathcal{P}_n$, we use $\mathcal{M}_{n-1}$ as the high-quality mesh for the non-rigid registration step, assuming that the motion performed between two consecutive frames is sufficiently small. Using these correspondences, we then deform

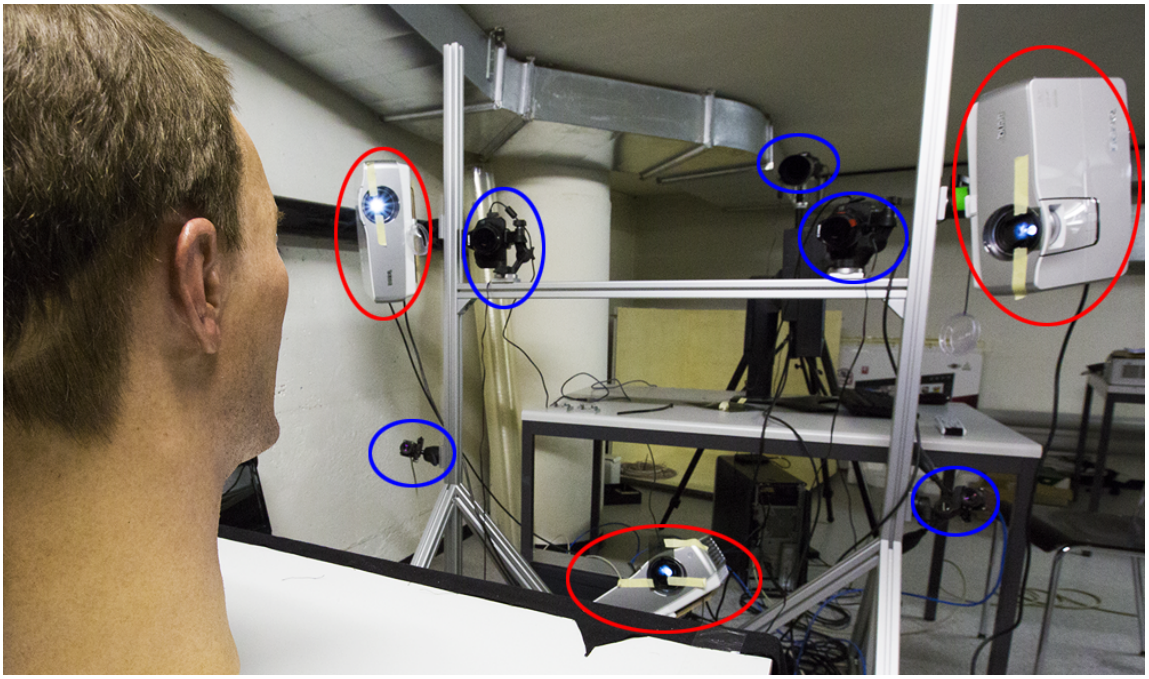


Figure 5.3: *Hardware setup: 5 cameras (blue) and 3 projectors (red) were used to reconstruct and illuminate the animatronic's face.*

$\mathcal{N}$ to obtain a deformed mesh $\mathcal{M}_n$ that matches $\mathcal{P}_n$ using linear rotation-invariant coordinates [Lip+05].

5.2.2 Actuator Control and Re-timing

We employ the physically based optimization method proposed by Bickel et al. [2012] to initially compute the animatronic actuation control. This method matches the deformation of the skin to each frame of the target sequence individually. As the animatronic head's range of motion is much more limited than the target performance, the resulting motion follows the target one as long as it can, and remains stationary once the target motion is out of range. Projecting the target sequence in such a case results in textures that continuously present motion while the animatronic avatar does not. In practice, this results in significant visual artifacts. We therefore suggest augmenting the actuation by taking dynamics into consideration, and not only the poses of the performance. Figure 5.4 exhibits a result of the process applied on the eyebrows-raising sequence. The graph shows how the resulting motion resembles the target one in terms of dynamics more than actual deformation, while the images illustrate the effects the process has on the performance itself.

Temporal optimization. As our actuated performance was created using physically based simulation and the mapping between actuation parameters and resulting skin deformation is non-linear, we chose to adapt the timing of the existing performance instead of creating a new one, assuming linear behavior only between adjacent frames. In other words, given a sequence consisting of N frames, we wish to create a new sequence of the same length, with each frame being a linear blend of two adjacent frames of the original motion. We start by analyzing the temporally coherent mesh sequence for the actuated performance, as described in Section 5.2.1, $\mathcal{M}_n, n = 1..N$, along with its correspondence to the target performance $\mathcal{T}_n, n = 1..N$. Denoting the re-timed mesh sequence as $\hat{\mathcal{M}}_n, n = 1..N$, we represent it by a vector $\tau \in [1..N]^N$ such that $\tau_n \in \tau$ defines $\hat{\mathcal{M}}_n = \mathcal{M}_{\lfloor \tau_n \rfloor} \cdot \alpha + \mathcal{M}_{\lceil \tau_n \rceil} \cdot (1 - \alpha), \alpha = (\tau_n - \lfloor \tau_n \rfloor)$. Using the error term discussed next, we wish to find a vector τ that minimizes the error between the target performance $\mathcal{T}_n$ and the augmented actuation frames $\hat{\mathcal{M}}_n$ induced by τ. In addition, we constrain τ to be temporally consistent such that each element $\tau_n \in \tau$ respects $\tau_n < \tau_{n+1}$. We employ constrained non-linear interior-point optimization to find the desired performance.

Error term. It has been shown that matching motion in the gradient space implies matching its dynamics instead of its pose and enhances realism [Seo+12]. However, as we do not have a linear face space, this principle is not directly applicable to our case. In the following we introduce an error term for the aforementioned optimization that is performance aware and helps avoiding local minima , by exploiting some key observations of our problem: First, each actuator drives the motion on a 1D curve. This means that instead of considering the 3D displacement of vertices, we can only consider their distance from the neutral pose. Second, target motion that resides within the avatar's range is reproduced fairly well, while large motion is clamped. Thus, considering the relative position (the ratio of every vertex's distance from the neutral pose to its maximum distance in the performance) describes the motion in a way that can be naturally translated to the avatar's gamut. Incorporating these observations and considerations, we get the following error term for a vertex v in a target performance mesh $\mathcal{T}_n$ and its corresponding position u in an actuated one $\hat{\mathcal{M}}_n$:

$$d(v,u) = |\mathbf{U}|\left(\frac{1}{|\mathbf{V}|}\frac{\partial|\mathbf{v}|}{\partial t} - \frac{1}{|\mathbf{U}|}\frac{\partial|\mathbf{u}|}{\partial t}\right)\cdot\omega_g + |\mathbf{U}|\left(\frac{|\mathbf{v}|}{|\mathbf{V}|} - \frac{|\mathbf{u}|}{|\mathbf{U}|}\right)\cdot\omega_s, \tag{5.1}$$

where $\mathbf{v}$ is the displacement of v from the neutral pose in the aforementioned frame, $\mathbf{V}$ is the maximum displacement of v in the whole sequence, and $\mathbf{u}$ and $\mathbf{U}$ are their counterparts in the actuated motion. We observed that

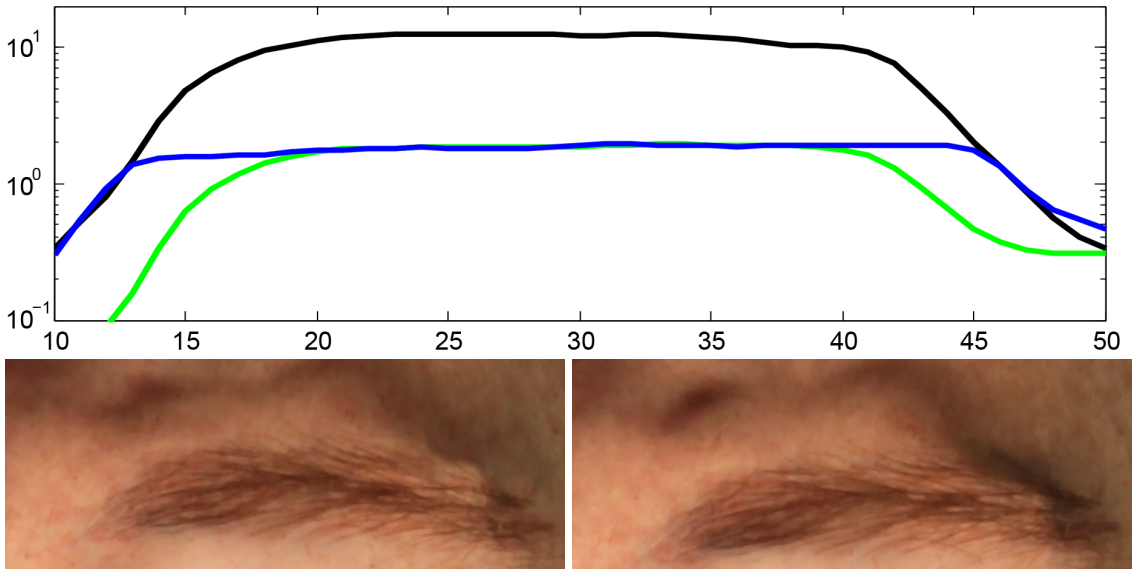


Figure 5.4: *Temporal remapping. Top: Graph showing the displacement of a vertex on the edge of an eyebrow. The original motion (black) surpasses the avatar's motion gamut. Static physical simulation matches the motion only within the gamut (blue). The remapped motion matches dynamic behavior instead (green). Bottom: The eyebrow position at frame 43. The projected features are nearly nonexistent while the eyebrow in the original motion still stays at peak position (left).*

adding the relative position error term prevents the solution from converging to a local minima. In our experiments we used the values of 0.85 and 0.15 for ω_g and ω_s, respectively.

Solution procedure. The optimization process starts with the initial guess that reproduces the original actuated motion $\tau = (1, 2, ..., N)$. During the optimization process, given the vector τ, we generate the induced actuated mesh sequence $\hat{\mathcal{M}}_n, n = 1..N$, and compute the aforementioned error term for a pre-selected random subset of the vertices. The error function used by the optimization $d : [1..N]^N \rightarrow \mathbb{R}$ is the Frobenius norm of the matrix containing all the error measures per vertex per frame. As this function is piecewise linear, its gradient can be computed analytically for each linear segment. To prevent local minima, we iteratively perturb the solution to generate new initial guesses by randomly sampling $\tau_n = [\tau_{n-1}, \tau_{n+1}]$ until there is no improvement of the solution in the current iteration. Finally, we replay the re-timed performance with the physical avatar and scan the exact geometry of $\hat{M}_n$ to obtain pixel-accurate data.

5.2.3 Detail Remapping

Having the re-timed avatar geometry, the next step is to map the details of the target performance to the avatar. The task of mapping one geometry to another is an ambiguous one, as some regions should be mapped to their semantic counterparts, such as the eyebrows in our case, while other regions, such as the lips, should deform freely to enhance expressiveness (see for example Figure 5.10). Therefore, we propose a method that does not alter geometry, but textures the avatar. This is done by rendering the performance from several points of view, deforming the rendered images to match the avatar according to user-specified semantics, and back-projects these images to the avatar while blending them in a confidence-driven manner.

Appearance transfer. Given a target performance sequence, consisting of N frames represented by a coherent set of meshes $\mathcal{T}_n, n = 1..N$, and a correlating sequence of the avatar $\hat{\mathcal{M}}_n$, the process starts with computing the correspondence between the neutral pose of the target performance, denoted by $\mathcal{T}_0$, and the neutral pose of the avatar $\mathcal{N}$. Using the method described in Section 5.2.1, the correspondence is achieved by registering $\mathcal{T}_0$ onto $\mathcal{N}$. Next, for every frame $\mathcal{T}_n$, we render it from m viewpoints, where $m = 4$ in our case. We carefully picked the views such that the complete facial area is covered. The result is a set of images $\mathbf{I}_i^{T_n}, i = 1..m$ and corresponding depth maps $\mathbf{Z}_i^{T_n}, i = 1..m$. As the avatar's meshes potentially cover more of the avatar itself than the target performance, we expand the target information of the rendered images $\mathbf{I}_i^{T_n}$ by mirroring the image across the mesh boundaries, adding a blurring term that grows with the distance from the boundary. While we achieved satisfactory results, in theory more sophisticated hole filling or texture generation algorithms could be used. Boundaries are determined by transitions between background and non-background depths in the depth maps $\mathbf{Z}_i^{T_n}$. The avatar's corresponding frame is also rendered, after being rigidly aligned with $\mathcal{T}_n$, creating the $\mathbf{I}_i^{\hat{M}_n}$ and $\mathbf{Z}_i^{\hat{M}_n}$ counterparts. Next, we deform the images $\mathbf{I}_i^{T_n}$ to match their avatar's counterparts, using moving least squares [SMW06]. The deformation is driven by a subset of vertices, which constrain the pixels they are projected to in $\mathbf{I}_i^{T_n}$ to move the projected position of their corresponding vertices in the avatar's rendering. Implicitly, this process deforms the low-frequency behavior of the target performance to match the avatar's one, while keeping true the high-frequency behavior of the target performance. The choice of the driving vertices is elaborated upon later in this section. Next, the images are projected back onto $\hat{\mathcal{M}}_n$, which means that every vertex receives the color from its rendered po-

sition on the deformed images, if it is not occluded. Blending between the different viewpoints is done based on the confidence of the vertex's color, determined by the cosine of the angle between the surface normal and viewing direction. As a final step, we perform for every vertex a few Laplacian temporal smoothing iterations on the resulting colors.

Conveying semantics. As aforementioned, the target performance is rendered and the images are deformed to match the physical avatar. The goal of the deformation is to adapt the target's features to the avatar while preserving the artistic intent of the performance. This notion suggests different behavior for different animations, and we allow the user to indicate the semantics of the animation by selecting individual or curves of vertices of the target performance and assign a property to it. These properties affect the behavior of the image deformation step described before. We have found that dividing the vertices into three types was sufficient to convey the semantics in our examples, and have used the same categorization for all of them. By default, all vertices are categorized as *free to move*, and have no effect on the image deformations. The second type, marked as *geometrical constraint*, enables the user to define vertices that will constrain the pixels that they are rendered to. The corresponding pixels of these vertices are moved to the position that their avatar's counterpart was rendered to, given that both are not occluded in the images. This type of constraint is usually used for vertices which are static throughout the performance, such as the nose, and is also useful for regions that should accurately match, such as the edges of the mouth and the eyebrows. The last type, marked as *view-dependent constraint*, relates to the fact that the geometries of the target performance and the avatar head do not match perfectly in some regions, and therefore the projection differs depending on the point of view. Marking these types of vertices with an associated viewpoint means that these vertices are constrained to match the avatar vertices they were projected closest to during the marked viewpoint. Figure 5.5 illustrates the effect of the different types of constraints: Removing the geometrical constraint from the eyebrows results in their projection on the middle of the forehead. Additionally, vertices that are marked as constrained with a front point of view are changed to a side one. This change proves unnatural from the front when the lip deviates away from the animatronic's geometry. In all our experiments, we have used 8 curves and 20 individual vertices that were geometrically constrained, and 2 curves and 5 individual vertices that were constrained view-dependent from the front view. Note that we have also experimented with different effect radii and also other types of constraints, such as snapping vertices back if they left the

Figure 5.5: *Semantics illustration. The marked vertices (left) are of the geometric type (blue) and view-dependent type (red). Removing constraints from the eyebrows results in an unnatural positioning (2^{nd} image) vs the original (3^{rd}). Vertices on the lips, marked with front view-dependent constraint (5^{th} image), are changed to a side one, which yields an unnatural look from the front (4^{th}).*

avatar's silhouette, but eventually found them unnecessary for our application.

5.3 Projection

After preprocessing the geometry and finally generating the per-vertex colors containing the desired shading, the model has to be projected accurately onto the physical avatar. This step involves rendering the geometry from the calibrated projector views and distorting the images to compensate for lens distortion. Additionally, we compute a light transport matrix that is used in a global optimization step for blending of multiple projector contributions, neutralizing physical light drop-off effects, and compensating for defocus and subsurface scattering to generate an optimized reproduction of high frequencies. To achieve this goal, besides the geometric calibration already described in Section 5.2, further data acquisition steps have to be carried out. Therefore the response curves of the used devices were linearized to simplify the image analysis and processing steps. While the camera response curves were linearized using the method described in [DM97], a Spyder4ELITE colorimeter was used for projector linearization and to match their color gamuts as well as lumen output. To match the cameras' color gamuts, an x-rite ColorChecker Classic based color transformation calibration was carried out.

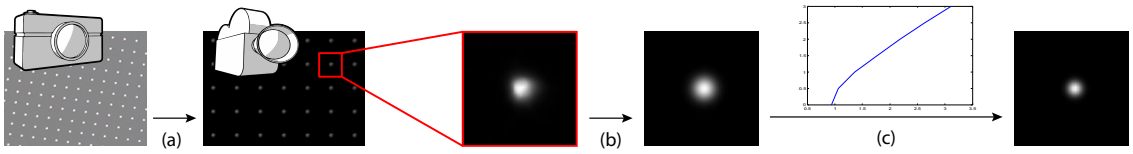


Figure 5.6: *Overview of the defocus measurement pipeline.* (a) *Back projection of the captured images to the projector's image plane and normalization.* (b) *Gauss fitting for each captured blob.* (c) *Recovering the amount of projector blur from the precomputed* LUT.

5.3.1 Defocus Data Acquisition

To accurately compensate for the projection defocus, the used PSF has to represent the physical defocus as precisely as possible. Following [NIS11; Ali+12], we approximate the projector defocus by a two-dimensional isotropic Gaussian function in the projector's image coordinate, depending on the pixel position and the distance to the projector:

$$PSF_z(xy, xy') = e^{-\frac{(x-x')^2+(y-y')^2}{\sigma^2_{x,y,z}}}. \tag{5.2}$$

Here, x and y are pixel coordinates of the pixel from which the projected light originates, x' and y' are the pixel coordinates of the target pixel that is illuminated by the defocused pixel, and z is the distance to the projector in world coordinates of the surface corresponding to the target pixel. σ is the standard deviation of the Gaussian function.

The PSF measurement process is based upon the one proposed in [NIS11]. The projector displays a two-dimensional grid of white pixels on black background onto a white, planar surface that is oriented to be orthogonal to the projection axis of the projector. This surface is placed at different distances around the focal plane of the projector and images are taken of the projected pixel pattern using one or more cameras. The Gaussian function is defined in the coordinate frame of the projector, requiring that all captured images be projected into the projector's image plane. Our implementation uses homographies [SSM01] for this purpose. Each back-projected image is split into patches, one for each projected pixel, and the PSF model is fitted to each patch, resulting in a σ value and a position x and y for each image patch. As our projectors did not exhibit significant chromatic aberrations, we captured only white patterns. In this case, the position (x and y) can be ignored, as any deviation of those coordinates from the coordinates of the originally projected pixel can be explained by inexact back projection. Using the computed homographies in combination with the geometrically calibrated cam-

eras and projectors, we also compute the distance to the projector for each pattern.

The σ values together with their respective distances and pixel coordinates constitute a dense, irregular field of defocus measurements, called a PSF field, that will be used to build the equation system for compensation. Depending on the density of the measurements, the defocus values for each point inside the covered volume can be interpolated with high accuracy. We observed that even while taking measures to reduce errors and minimize the influence of noise and environment light, the proposed measurement procedure produces σ values much greater than 0, even when measuring next to the focal plane. In our setup the minimal σ values were around 0.8. As our PSF model describes Gaussian functions in the projector image space, a σ value of 0.8 translates into a Gaussian that includes already severe defocus, covering multiple neighboring pixels. Reasons for this additional defocus include coma and chromatic aberrations of the camera lenses, its aperture settings, sampling inaccuracies both on the camera CCD and during the back projection step, and noise.

We propose an additional calibration step, referred to as sigma calibration, designed to uncover the blurring behavior of the capturing and model fitting pipeline. For this, we place the same white plane that was used for the measurements above into the focal plane and project a single pixel on a black background, followed by Gaussian blurred versions of the same with increasing σ. The captured patterns were again fitted to Gaussians, which results in a lookup table (LUT) between the σ values of the actually projected Gaussian functions and the ones found using the measurement pipeline. The overall process is illustrated in Figure 5.6.

Besides measurement of the projector defocus, subsurface scattering is measured and modeled as well. The modeling was done using the method described in [DI11] while the measurement was carried out using a device based on [Wey+06].

5.3.2 Projection Image Computation

To optimize the projected images, the light transport is computed and compensated for. We modeled the light transport as matrix-vector multiplication:

$$C = LP, \tag{5.3}$$

where P is a vector containing the projected images, L is a matrix containing the light transport, and C is the output of the system. The semantic mean-

ing of C depends on what aspect of the projection system is of interest. In previous works in the context of light transport and defocus compensation [ZN06; Ali+12; WB07], C corresponds to an image captured by a designated camera that is used as a proxy for a human observer, and L encodes the light transport from one or more projectors to this camera.

To the best of our knowledge, we present the first work on pre-correcting defocus compensation for multi-projector systems that does not use a reference camera as optimization target. Instead, we completely work in the image planes of the involved projectors, treating them as virtual cameras. In this case, C represents the set of images that would be captured by the projectors. As the compensation images are generated using the parameters stored directly for each projector pixel, the resulting compensation is independent of the camera viewpoint and thus is not influenced by occlusions, obliqueness, camera defocus, etc., which would occur from almost any camera viewing position.

Compensation of the light transport, i.e. finding the images P that produce the output C when being projected, conceptually involves an inversion of the light transport: $P' = L^{-1}C'$. Here C' is the desired output of the system and P' is the input that produces it when projected. In most cases, directly inverting L is impossible because L is not full rank. As was done in [ZN06] and [Ali+12], we instead reformulate the compensation as a minimization problem: $P' = \text{argmin}_{0 \leq P \leq 1} \|LP - C'\|^2$. In the course of building up the components of the equation system, this minimization will be extended to contain locally varying upper bounds, weighting of individual pixels, and additional smoothness constraints, resulting in the following minimization:

$$P' = \underset{0 \leq P \leq U}{\text{argmin}} \|W\,(TP - S)\|^2 = \underset{0 \leq P \leq U}{\text{argmin}} \|W \left(\begin{bmatrix} L \\ Smooth \end{bmatrix} P - \begin{bmatrix} C \\ 0 \end{bmatrix} \right) \|^2.$$

S is a vector containing the target images C' and the smoothing target values of constant 0. T is a matrix consisting of the light transport L and the smoothing terms $Smooth$. W is a diagonal matrix containing weights for each equation. Finally, U contains the upper bounds of the projected image pixel values.

Light Transport. Below, we build up the light transport iteratively by its components. For projector defocus, σ is looked up in the PSF field at the pixel coordinates of the source pixel as well as at the depth of the target pixel. The PSF model is then evaluated using σ, and the resulting value is normalized such that all the light emitted at the same source pixel sums up to 1. To ensure that the compensated pictures result in a uniformly bright

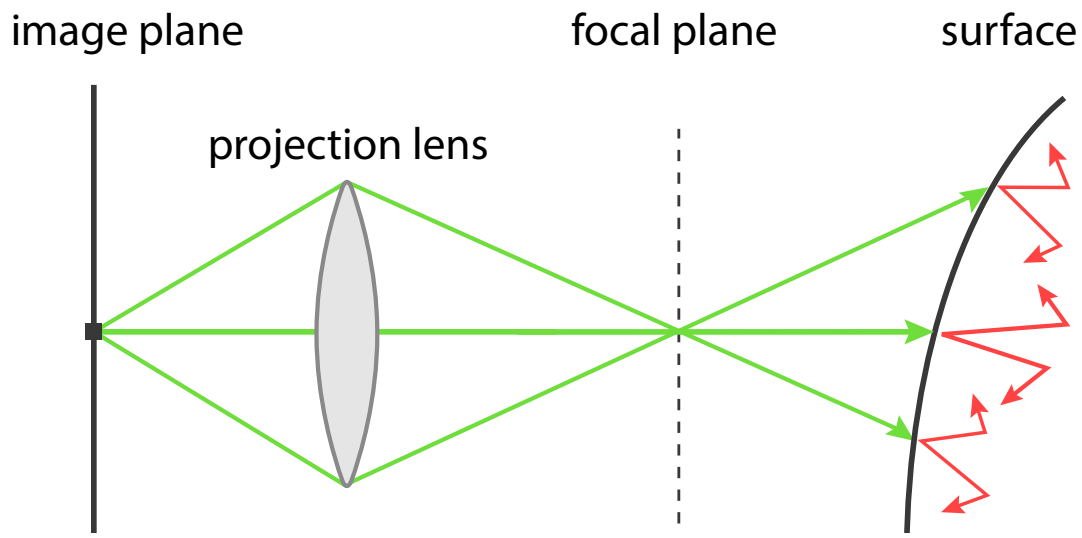


Figure 5.7: *Simplified visualization of the spatial distinction between projector defocus resulting from its lens properties (green) and subsurface scattering (red). The defocus originates before the light physically reaches the surface, while subsurface-scattering evolves only once it has hit the surface.*

appearance, light drop-off caused by distance to the projector and the incidence angle of the light at the surface is included in the light transport. This is done by multiplying the light drop-off factor on top of the defocused projection computed previously.

As illustrated in Figure 5.7, subsurface scattering physically happens after projector defocus. Thus it is possible that light emitted from one pixel can travel to the same target pixel using multiple paths, so care has to be taken to sum up those contributions correctly. The subsurface scattering factor is looked up in the previously measured scattering profile with the world coordinate distance between the two involved surface points. This formulation is not quite correct, as these measurements are valid only for flat patches of silicone with a certain thickness. General surfaces are neither of uniform thickness nor flat, however, and especially in concave parts, the point distance in world coordinates does not correspond to the distance on the surface. But these inaccuracies are relatively small and don't carry much weight when compared to other sources of errors, such as inexact geometry and calibrations. As such, we do not handle these effects.

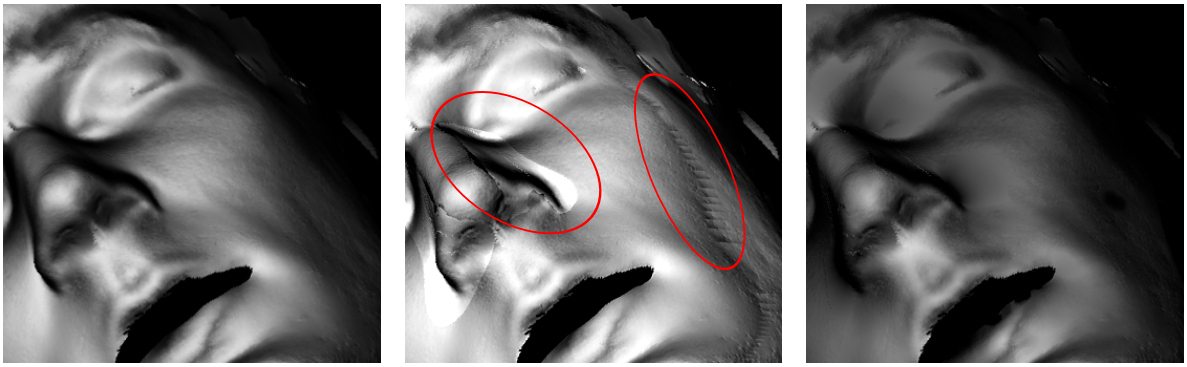
This finishes the single projector light transport (PLT). The following modifications are needed only in multi-projector systems; they fill in the cross PLT without changing the already computed values. Instead of recomputing projector defocus and subsurface scattering for the cross PLT, the rele-

vant values are looked up in the results of the single PLT using a projective mapping between the projectors. See the appendix for a description of this lookup process. To make sure that the computed cross PLT actually deals in correct units, the relative brightness of the involved projectors has to be considered as well. We use three projectors of the same make and model, and have calibrated them to be of the same brightness as part of the projector response curve linearization mentioned earlier.

Blending Multiple Contributions. In multi-projection systems, blending maps are applied to ensure consistent intensities in overlapping projection areas (cf. e.g. [Ras+98; Har+06]). This is especially important when projecting onto objects that are discontinuous when seen from a specific projector. We use a geometry-based blending map calculation approach using shadow volumes to detect discontinuous regions in the projector image planes and smoothly fade out the individual projector intensities in these areas as well as at the edges of the image planes in overlapping areas.

Previous work on multi-projector defocus compensation, such as [Ali+12], does not take blending into account. This can be a serious shortcoming, as it produces noticeable artifacts in the presence of discontinuities. Not involving blending maps while at the same time compensating for light drop-off caused by incidence angle has the effect that projectors increase their intensity when projecting onto oblique surfaces, instead of leaving the illumination of such surfaces to another projector in a more suitable position.

We propose to include the blending maps into the minimization as upper bounds (U in equation 5.4). See Figure 5.8 for a comparison of compensation results with and without blending. These results were computed for a three-projector system (see Figure 5.3), and the compensation images of the lower projector are shown. It can be seen that the result without blending (b) contains severe artifacts. They are most noticeable in areas of discontinuities such as around the nose and on the cheeks. Applying the proposed approach reduces the artifacts below a perceptual level (c). In regions where projectors overlap, one point on the target surface is represented by multiple pixels in the image planes of multiple projectors. If each of those pixels had the same weighting in the residual computation, overlapping regions would be treated as more important than non-overlapping regions. Not all solution pixels have the same accuracy requirements: It is more important for each projector to find good solutions for image patches for which it is the only projector, or onto which it projects orthogonally. These criteria are also followed when constructing blending maps, which makes blending maps good weights for the individual equations in the system (W in equation 5.4).



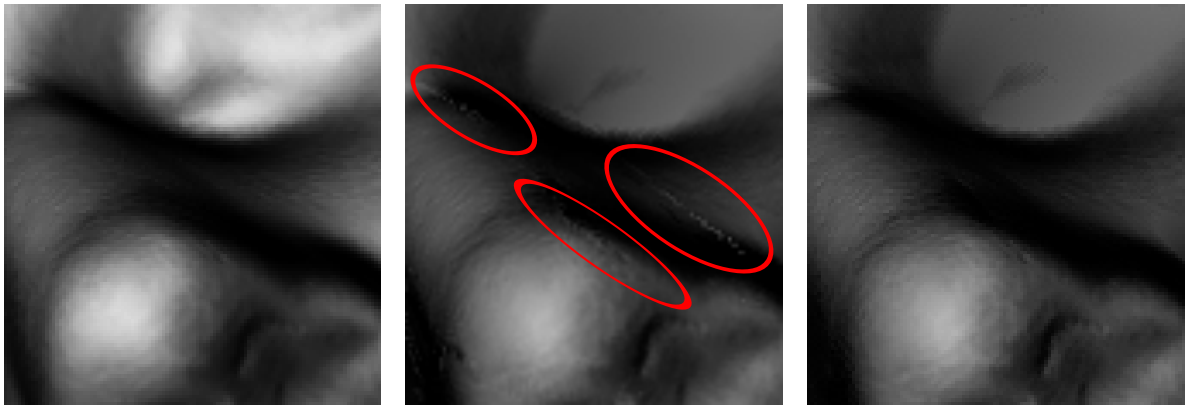
(a) (b) (c)

Figure 5.8: *Blending Comparison.* (a) *Input image.* (b) *Compensation image without blending; note the marked artifacts.* (c) *Compensation image with blending maps as upper bounds.*

See Figure 5.9 for a comparison of compensation results with and without weighting. These images show an excerpt around the nose of the same three-projector system as before. (b) was computed with blending maps as upper bounds, but without weighting the equations. Note the artifacts (in red) that disappear when including the weights, resulting in (c).

Smoothing. Even careful PSF measurement and sigma calibration might lead to a slight under- or overestimation of the projector defocus, resulting in visible artifacts caused by the projection of incorrect compensation images. Additionally, in regions where multiple projectors overlap, there is no guarantee in which way the compensation image is composed. This can lead to the case that for two neighboring pixels, one pixel is completely produced by the first projector and the other by the second projector. In this case, small calibration errors will become immediately apparent. Both of these issues can be reduced by introducing additional smoothness constraints. We implemented smoothness constraints similar to the ones proposed in [Ali+12]. We refer to the supplemental material for a description of the smoothness constraints.

Solving. In our implementation, we used the iterative, constrained, steepest descent algorithm presented in [ZN06] as a solver for the equation system. See the supplemental material for a description of how to deal with the global scaling of the system.



(a) (b) (c)

Figure 5.9: *Weighting comparison. (Close-up of the image shown in Figure 5.8)* (a) *Input image, showing the nose.* (b) *Compensation image with blending maps as upper bounds but no weighting, leading to artifacts (red).* (c) *Adding the blending maps as weights removes those errors.*

5.4 Results and Discussion

To evaluate the performance of our projection-based enhancements, we used the silicone animatronic head described by Bickel et al. [2012] and mapped a performance capture sequence of a real actor onto it. In addition, some of the input sequences were artistically altered to simulate a man older than the one who actually performed. Figure 5.10 shows different results of our proposed method. As can be seen, the actuators of the animatronic are not able to generate the complex skin deformation required to accurately reproduce the input geometry. Adding the missing information using our proposed projection mapping significantly enhances the high-frequency components, and thus the expressiveness of the performance. As the process is designed to optimize for several viewing angles, we demonstrate the robustness to viewer positions by using a hand-held camera, with and without ambient lighting in the room, in the accompanying video and in Figure 5.11. Furthermore, to emphasize the effect of our two-scale approach, we keep the animatronic head static, and perform the suggested method for one of the sequences. As can be seen in the accompanying video, as well as in Figure 5.12, while an illusion of the desired performance can be generated using only the projection or only physical animation, the combination of the two produces a far more compelling result.

To evaluate the quality improvement of our multi-projector optimization method, we used the structural similarity index (SSIM) [Wan+04b], which is a method for assessing the perceptual quality of a distorted image when

Figure 5.10: *Three captured results of extreme poses, generated with our system. For every frame, the desired appearance is on the left, the robot configuration under uniform white illumination is in the middle, while the augmented result is on the right.*

Figure 5.11: *Results of a single frame captured from random viewing angles, illuminated only by projectors (top row) and with ambient lighting in the room (bottom row).*

Figure 5.12: *The neutral pose (left) of a sequence, compared to an extreme pose while the animatronic head is kept static (middle), and while it is actuated according to the proposed method (right). The head configuration under uniform illumination, and the target appearances are shown to the left of each result.*

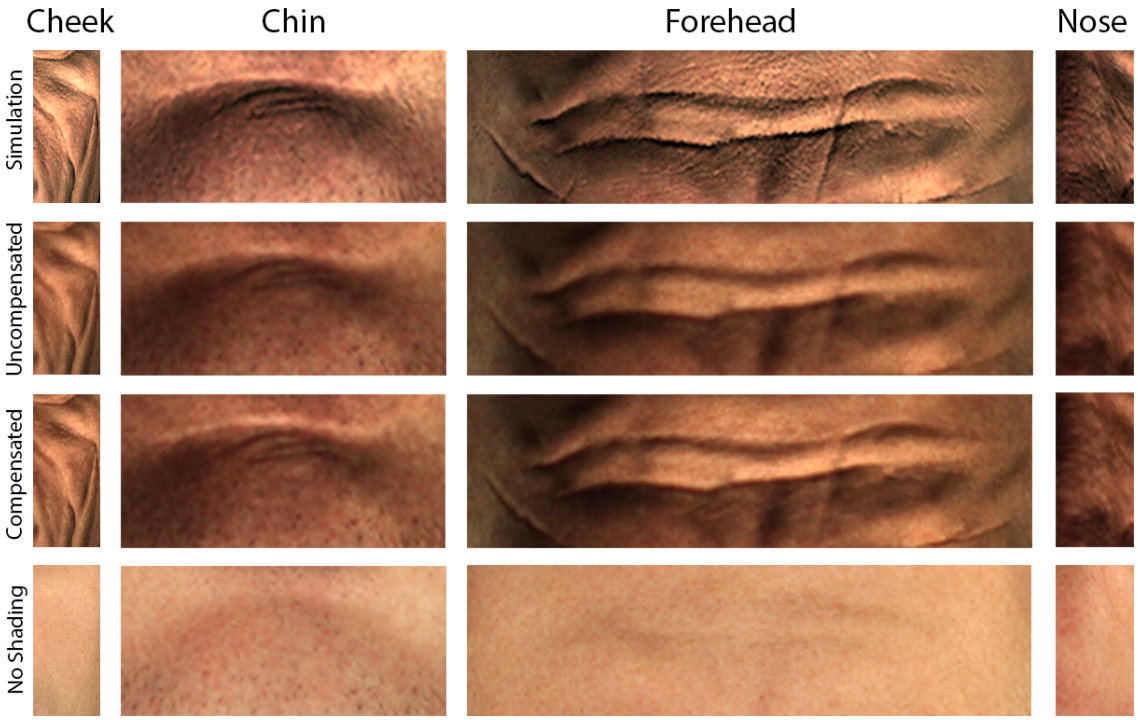


Figure 5.13: *Close-up comparison of the projection shown in Figure 5.1. Upper row: Simulations of the desired appearance. Second row: Uncompensated results. Third row: the compensated results. Fourth row: Appearance of the head when no shading is projected.*

Area	No Shading	Uncompensated	Compensated
Cheek	0.543	0.816	0.864
Chin	0.722	0.889	0.908
Forehead	0.632	0.815	0.849
Nose	0.669	0.864	0.885

Table 5.1: *SSIM evaluation results for the cropped image regions shown in Figure 5.13.*

compared to the original. We used a modified version of SSIM to compare the projection results of uncompensated and compensated shadings to a ground truth image. This was generated by rendering the input image from a calibrated camera and using the color mapping technique described in [Gru13] in an inverse manner to simulate the per-pixel surface color modulations. It has been modified in that it does not take the absolute pixel values into account, but only compares contrast and structure. This measure results in a value between -1 and 1 where 1 corresponds to no distortion.

Figure 5.13 shows the excerpts from the final frame of the growing-old sequence shown in Figure 5.1. The uncompensated and the compensated projections were compared to the simulated ground truth, resulting in the SSIM scores contained in Table 5.1. As can be seen, besides the perceived improvements presented in Figure 5.13, the defocus compensation results in a measurable increase in similarity to the simulated ground truth. As the finite pixel resolution as well as the subsurface scattering properties of the silicone skin constrain the reproduction quality of the input shading, a value of 1.0 is impossible to achieve with the presented setup. In practice we found compensating for subsurface scattering to be more important than for defocus. Experiments showed that for the silicon head, the subsurface scattering compensation enhanced the image quality significantly more when compared to defocus compensation alone, as it also reduces image contrast in well-focused areas.

On our machine featuring a quad-core i7 Intel CPU, 24 GB of RAM, and an NVidia QuadroPlex graphics card, the creation of a detailed mesh out of the acquired point cloud lasts about 3 minutes per frame, and the application of a target frame onto it takes about 1 minute. The actual projection image generation is performed in real-time. The computation of a light transport matrix for one pose takes about 11 minutes, while the compensation of a set of projection images takes about 8 minutes.

5.5 Summary and Future Work

In this paper we presented a novel approach using spatially varying illumination to enhance the appearance and expressiveness of a silicone-skin-based head animatronic. We demonstrated that a carefully calibrated multi-projector system in combination with geometrical mapping can significantly enhance its realism by projecting high-frequency skin structures that cannot be reproduced by the animatronic's actuators and the silicone skin alone.

In the future, we are planning to integrate the proposed approach into a real-time, live feedback system to enable a realistic and responsive animatronic interaction. While the software tools for real-time geometry mapping are already available, this step requires a sophisticated engineering effort in terms of accurate hardware setup and synchronization. The used subsurface scattering compensation uses a simplified, spatially uniform description of the subsurface scattering behavior. While this is the result of a missing measurement device, a future research direction would be the utilization of the projector-camera system to acquire accurate, spatially varying subsurface scattering information similar to the work presented in [GD08]. Another related future research direction would be an accurate estimation of the spatially varying surface BRDF to also enable a view-independent photometric projector compensation.

CHAPTER 6

Conclusion

In this chapter, we conclude this dissertation by summarizing the major contributions and discussing future research directions.

6.1 Contributions

In this thesis, we have presented the high-level process of digital facial content creation - capture, augmentation and display - and have proposed an advancement to each of these steps. In all chapters, we have focused on delivering content in the highest level of realism aiming to push the envelope with regards to crossing the uncanny valley. Our work potentially reduces the manual labor required to produce production level facial content by a great deal, enabling the creation of substantially more content.

We first address the capturing step, where we have realized that humans identify emotions by primarily using the eye region, but despite the important role of this region, existing methods are unable to provide the level of geometric detail and motion required for production level content. More specifically, acquiring eyelids is very challenging due to extreme deformations while the eye opens, stretching over the eyeball when the eye is shut and substantial occlusions due to concavities and eyelashes. We therefore propose in Chapter 3 the first method for detailed spatio-temporal reconstruction of eyelids. Our approach combines a specially tailored geometric deformation model with image data, leveraging multi-view stereo, optical flow, contour tracking and wrinkle detection from local skin appearance.

Our deformation model is anatomically motivated and is designed to produce plausible eyelid motions. Our results demonstrate that the model is able to provide a high-resolution mesh that deforms over time, reflecting detailed dynamic skin features even for regions that are occluded or undergo extreme deformations. Since this approach does not rely on the capture approach, it can be easily integrated into any performance capture pipeline, be it passive or active, that records sufficiently high-resolution footage of the eye region. As we demonstrate with several results, our system allows the reconstruction of an expressive, dynamic model of the eye region at a quality level that has never before been possible, increasing the fidelity of this very important region.

In Chapter 4, we propose an enhancement to the second step of the digital facial content creation pipeline - augmentation. While many methods exist for art-directable facial animations which can easily be tuned for a desired performance, they typically lack the subtle spatial and temporal details that make a facial performance compelling. Of course, as seen in Chapter 3, high-resolution capture devices exist, which can acquire an expressive facial performance, however the result can only be played back and is not directable. Therefore, we propose a data-driven approach to enhance the expressiveness of facial geometry and motion. Empolying a high fidelity facial performance scanner, we record an individual exploring the full range of facial expressiveness. A model is then built, which can be used to automatically transfer subtle spatial and temporal features to lower-resolution facial animations that lack expressive details, significantly increasing the perceived realism. As mentioned, our system also takes advantage of the timing information in these recordings to enhance facial keyframe interpolation reflecting the nonlinearities of a real actor's performance. We demonstrate the robustness of our approach by enhancing a variety of input animations, including hand-animated facial rigs, face models driven by low-resolution motion capture data, morphable models animated using video data, and performance reconstructions generated with a Kinect. We also show that our algorithm outperforms the current state-of-the-art approach for data-driven facial performance synthesis [Ma+08].

Lastly, in Chapter 5 we also propose an enhancement to the unavoidable step of any digital content creation pipeline - display. Here we aim to bring characters to life by bridging the gap between high-quality digital content and physical avatars who convey a much greater sense of presence. Unlike digital face models, animatronic figures or robotic avatars are unable to reproduce detailed facial motions due to physical constraints. In Chapter 5 we significantly increase the expressiveness of such figures, by adding additional degrees of freedom with projector based illumination, effectively con-

trolling their appearance in a digital way. We present a two-scale model for controlling facial appearance, tailored to animatronic heads. Low-frequency motions that can be reproduced by the physical head are represented as control parameters of actuators. High-frequency details and motions that are outside of the physical motion gamut are added through illumination, similar to the commonly practiced texturing concept. We also propose an efficient spatio-temporal method for decomposing the motion in gradient space, enabling us to actuate the avatar in a way that matches the desired appearance but also the desired timings, maintaining the correlation of low-frequency physical motion and high-frequency details appearance. Furthermore, we present a complete multi-camera and -projector system, allowing efficient optimization of the projection quality in terms of focus and contrast. Defocused projections and subsurface scattering in practice reduce contrast and impair sharpness. We present a model-based multi-projector optimization step to alleviate these problems through considerations of physical light drop-off, smooth blending in overlapping regions, projection defocus, and subsurface-scattering.

6.2 Future Work

In this thesis we have proposed tools to address different problems that arise during facial content creation. This final Section outlines some areas of future work in the context of the solutions presented in the individual thesis Chapters. We end with a more global vision of future directions and developments.

Eyelids Reconstruction We have presented a system to reconstruct eyelids with spatio-temporal details. However, this pipeline is not fully automatic and relies on a few manual steps, initialization and parameter tuning. We believe that all of these steps can be fully automatic in the future. Our reconstruction relies on the wrinkle probability map, which currently employs general purpose kernel. A more sophisticated means of computing and extracting the wrinkles would an interesting direction to look at as well. Some expressions such as extreme grinning or squinting typically induce wrinkle formations that our model does not handle. In the future, we would like to extend our model to handle more general cases. Furthermore, as we compute several data terms, such as the eyelid contours, relative to the front camera, we can only handle minor head rotations. Extending the method to allow for large head rotations could be an interesting avenue for future research. Finally, our system focuses solely on performance capture. For

future work, an interesting avenue would be to add animation control, enabling easy creation of eyelids for facial rigs, and performance and detail transfer of to different characters, thereby bringing the expressiveness of virtual characters to a new level.

Performance Enhancement Our performance enhancement system is able to augment low-resolution input animations with high frequency compelling details. However, an interesting area for future research could be to analyze and correct low-frequency errors. In this work we assume that the low-frequency component of the input animation is accurate, matching the actor's facial expressions and motion. Correcting noisy input, impossible configurations and dynamics would greatly benefit such systems. Furthermore, another interesting direction would be to explore performance transfer, using input animations from one actor with a database from another.

Projection Based Augmentation In Chapter 5, we have presented a system to display target animations of a robotic avatar head. In the future, integrating the proposed approach into a real-time, live feedback system would greatly enhance the system. This could enable a realistic and responsive animatronic interaction, with both entertainment and tele-presence applications. The appearance estimation incorporate in the system is also rather simplified. As future work, a more accurate and spatially varying representation of sub-surface scattering behavior and BRDF would be interesting to define and measure.

Outlook The work presented in this thesis could be an important step in several interesting directions. For example, specially tailored deformation models such as the one presented in Chapter 3 could be applied to other soft tissue capturing applications, such as skin on the hand and fingers, elbows etc. The performance enhancement system presented in Chapter 4, if became real-time and retargetable, could lead to a new kind of facial puppetry, with applications spanning through entertainment, tele-presence or even impersonation. Combined with the work suggested in Chapter 5, the possible applications for powerful and live facial augmentation are limitless, from using the human body as a display, through virtual makeup, versatile performances and much more.

To conclude, this thesis presents advancements to the facial content creation pipeline, in the fields of capturing, authoring and display. We believe that this work constitutes another step towards crossing the uncanny valley, and

hope that this work would inspire facial animation research and be used to help it progress towards the exciting avenues it holds for us in the future.

References

[AA00] E. D. Andersen and K. D. Andersen. "The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm". In: *High Performance Optimization*. 2000, pp. 197–232.

[Ale+10] Oleg Alexander et al. "The Digital Emily Project: Achieving a Photoreal Digital Actor". In: *IEEE Computer Graphics and Applications* 30.4 (2010), pp. 20–31.

[Ale+13] Oleg Alexander et al. "Digital Ira: creating a real-time photoreal digital actor". In: *ACM SIGGRAPH 2013 Posters*. 2013.

[Ali+12] Daniel G. Aliaga et al. "Fast high-resolution appearance editing using superimposed projections". In: *ACM Trans. Graph.* 31.2 (2012), 13:1–13:13. ISSN: 0730-0301.

[Bar+09] Ilya Baran et al. "Semantic Deformation Transfer". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 28.3 (2009), 36:1–36:6.

[BB14] Thabo Beeler and Derek Bradley. "Rigid Stabilization of Facial Expressions". In: *ACM Trans. Graph.* 33.4 (2014), 44:1–44:9. ISSN: 0730-0301.

[BCS97] Christoph Bregler, Michele Covell, and Malcolm Slaney. "Video Rewrite: Driving Visual Speech with Audio". In: *Proc. SIGGRAPH 97*. 1997, pp. 353–360.

[BE06] Oliver Bimber and Andreas Emmerling. "Multifocal Projection: A Multiprojector Technique for Increasing Focal Depth". In: *Trans. Visualization and Computer Graphics* 12.4 (2006), pp. 658–667. ISSN: 1077-2626.

[Bee+10] T. Beeler et al. "High-Quality Single-Shot Capture of Facial Geometry". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 29 (4 2010), 40:1–40:9.

[Bee+11] Thabo Beeler et al. "High-quality passive facial performance capture using anchor frames". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 30 (4 2011), 75:1–75:10.

[Bee+12] Thabo Beeler et al. "Coupled 3D Reconstruction of Sparse Facial Hair and Skin". In: *ACM Trans. Graph.* 31.4 (2012), 117:1–117:10. ISSN: 0730-0301.

[Ben+07] P. Benzie et al. "A Survey of 3DTV Displays: Techniques and Technologies". In: *Circuits and Systems for Video Technology, IEEE Transactions on* 17.11 (2007), pp. 1647–1658.

[Ber+14b] Amit H. Bermano et al. "Facial Performance Enhancement Using Dynamic Shape Space Analysis". In: *ACM Trans. Graphics* 33.2 (2014).

[Bic+07] Bernd Bickel et al. "Multi-scale capture of facial geometry and motion". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 26.3 (2007), p. 33.

[Bic+08a] Bernd Bickel et al. "Pose-Space Animation and Transfer of Facial Details". In: *Proc. SCA*. 2008, pp. 57–66.

[Bic+08b] Bernd Bickel et al. "Pose-Space Animation and Transfer of Facial Details". In: *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. 2008, pp. 57–66.

[Bic+12] Bernd Bickel et al. "Physical face cloning". In: *ACM Trans. Graph.* 31.4 (2012).

[Bim+07] Oliver Bimber et al. "The Visual Computing of Projector-Camera Systems". In: *Proc. Eurographics (State-of-the-Art Report)*. 2007, pp. 23–46.

[BKN02] Yosuke Bando, Takaaki Kuratate, and Tomoyuki Nishita. "A Simple Method for Modeling Wrinkles on Human Skin". In: *Proc. Pacific Graphics*. 2002.

[Bla+03] V. Blanz et al. "Reanimating Faces in Images and Video". In: *Computer Graphics Forum* 22.3 (2003), pp. 641–650.

[Bor+03] G. Borshukov et al. "Universal Capture – Image-based Facial Animation for "The Matrix Reloaded"". In: *ACM SIGGRAPH 2003 Sketches & Applications*. 2003.

[Bot+06] Mario Botsch et al. "Deformation Transfer for Detail-Preserving Surface Editing". In: *Vision, Modeling & Visualization*. 2006, pp. 357–364.

[BPG04] F. Blais, M. Picard, and G. Godin. "Accurate 3D acquisition of freely moving objects". In: *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. 2004, pp. 422–429.

[Bra+08] D. Bradley et al. "Markerless Garment Capture". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* (2008), p. 99.

[Bra+10] D. Bradley et al. "High Resolution Passive Facial Performance Capture". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 29 (4 2010), 41:1–41:10.

[Bra99] M. Brand. "Voice puppetry". In: *Proc. SIGGRAPH 99*. 1999, pp. 21–28.

[Bro+04] Thomas Brox et al. "High accuracy optical flow estimation based on a theory for warping". In: *ECCV*. Springer, 2004, pp. 25–36.

[BS08] Mario Botsch and Olga Sorkine. "On linear variational surface deformation methods". In: *Visualization and Computer Graphics, IEEE Transactions on* 14.1 (2008), pp. 213–230.

[Buc+00] Ian Buck et al. "Performance-Driven Hand-Drawn Animation". In: *NPAR 2000 : First International Symposium on Non Photorealistic Animation and Rendering*. 2000, pp. 101–108.

[BV99] Volker Blanz and Thomas Vetter. "A morphable model for the synthesis of 3D faces". In: *Proc. Computer graphics and interactive techniques*. 1999, pp. 187–194.

[BWP13] Sofien Bouaziz, Yangang Wang, and Mark Pauly. "Online modeling for realtime facial animation". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 32.4 (2013), 40:1–40:10.

[BZK85] Andrew Blake, Andrew Zisserman, and Greg Knowles. "Surface descriptions from stereo and shading". In: *Image and Vision Computing* 3.4 (1985), pp. 183–191.

[Bér+14] Pascal Bérard et al. "High-quality Capture of Eyes". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 33.6 (2014).

[Bü+05] Bernhard Büttgen et al. "CCD/CMOS Lock-in pixel for range imaging: challenges, limitations and state-of-the-art". In: *In Proceedings of 1st Range Imaging Research Day*. 2005, pp. 21–32.

[Can86] John Canny. "A computational approach to edge detection". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6 (1986), pp. 679–698.

[Cao+04] Y. Cao et al. "Real-time speech motion synthesis from recorded motions". In: *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. 2004, pp. 345–353.

[Cao+12] Xudong Cao et al. "Face alignment by Explicit Shape Regression". In: *IEEE CVPR*. 2012, pp. 2887–2894.

[Cao+13] Chen Cao et al. "3D shape regression for real-time facial animation". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 32.4 (2013), 41:1–41:10.

[CB02] Erika Chuang and Chris Bregler. *Performance driven facial animation using blendshape interpolation*. Tech. rep. CS-TR-2002-02. Department of Computer Science, Stanford University, 2002.

[CET01] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. "Active appearance models". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.6 (2001), pp. 681–685.

[CH07] Jinxiang Chai and Jessica K. Hodgins. "Constraint-Based Motion Optimization Using a Statistical Dynamic Model". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 26.3 (2007), 8:1–8:9.

[CHZ14] Chen Cao, Qiming Hou, and Kun Zhou. "Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 33.4 (2014), 43:1–43:10.

[CP99] Sagar M. Charette P. "The Jester, http://www.pactitle.com/". In: *the Electronic Theater at Siggraph*. 1999.

[CXH03] Jin-Xiang Chai, Jing Xiao, and Jessica Hodgins. "Vision-based control of 3D facial animation". In: *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. 2003, pp. 193–206.

[Dal+11] Kevin Dale et al. "Video face replacement". In: *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 30.6 (2011), 130:1–130:10.

[Deb+00] Paul Debevec et al. "Acquiring the reflectance field of a human face". In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '00. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 145–156. ISBN: 1-58113-208-5.

[Des+99] Mathieu Desbrun et al. "Implicit fairing of irregular meshes using diffusion and curvature flow". In: *Proc. SIGGRAPH 99*. 1999, pp. 317–324. ISBN: 0-201-48560-5.

[DI11] Eugene D'Eon and Geoffrey Irving. "A quantized-diffusion model for rendering translucent materials". In: *ACM Trans. Graph.* Vol. 30. 4. 2011, p. 56.

[dLE07] Eugene d'Eon, David Luebke, and Eric Enderton. "A system for efficient rendering of human skin". In: *ACM SIGGRAPH 2007 sketches*. SIGGRAPH '07. San Diego, California: ACM, 2007.

[DLN05a] Z. Deng, J.P. Lewis, and U. Neumann. "Synthesizing speech animation by learning compact speech co-articulation models". In: *Computer Graphics International (CGI)* (2005), pp. 19–25.

[DLN05b] Zhigang Deng, J.P. Lewis, and U. Neumann. "Automated eye motion using texture synthesis". In: *CGA* 25.2 (2005).

[DM96] Douglas DeCarlo and Dimitris Metaxas. "The integration of optical flow and deformable models with applications to human face shape and motion estimation". In: *CVPR*. 1996, pp. 231–238.

[DM97] Paul E. Debevec and Jitendra Malik. "Recovering high dynamic range radiance maps from photographs". In: *Proc. of ACM SIGGRAPH*. 1997, pp. 369–378.

[DMB11] Ludovic Dutreve, Alexandre Meyer, and Sada Bouakaz. "Easy Acquisition and Real-time Animation of Facial Wrinkles". In: *Comput. Animat. Virtual Worlds* 22.2-3 (2011), pp. 169–176.

[EF77] Paul Ekman and Wallace V Friesen. "Facial action coding system". In: (1977).

[EF78] P. Ekman and W.V. Friesen. "The Facial Action Coding System: A Technique for the Measurement of Facial Movement". In: *Consulting Psychologists*. 1978.

[EGP02] T. Ezzat, G. Geiger, and T. Poggio. "Trainable videorealistic speech animation". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 21.3 (2002), pp. 388–398.

[Ekm+93] Paul Ekman et al. "Final report to NSF of the planning workshop on facial expression understanding". In: *Human Interaction Laboratory, University of California, San Francisco* 378 (1993).

[Ess+96] Irfan Essa et al. "Modeling, Tracking and Interactive Animation of Faces and Heads: Using Input from Video". In: *Computer Animation '96*. 1996, pp. 68–79.

[FKY08] Wei-Wen Feng, Byung-Uck Kim, and Yizhou Yu. "Real-Time Data-Driven Deformation Using Kernel Canonical Correlation Analysis". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 27.3 (2008), 91:1–91:9.

[Fra+09] Guillaume François et al. "Image-based modeling of the human eye". In: *IEEE TVCG* 15.5 (2009), pp. 815–827.

[FYK10] Wei-Wen Feng, Yizhou Yu, and Byung-Uck Kim. "A Deformation Transformer for Real-time Cloth Animation". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 29.4 (2010), 108:1–108:9.

[Gar+13] Pablo Garrido et al. "Reconstructing Detailed Dynamic Face Geometry from Monocular Video". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*. Vol. 32. 6. 2013, 158:1–158:10.

[GD08] Abhijeet Ghosh and Paul Debevec. "Estimating multi-layer scattering in faces using direct-indirect separation". In: *ACM SIGGRAPH 2008 talks*. SIGGRAPH '08. Los Angeles, California, 2008, 2:1–2:1. ISBN: 978-1-60558-343-3.

[Gel08] T. Geller. "Overcoming the Uncanny Valley". In: *Computer Graphics and Applications, IEEE* 28.4 (2008), pp. 11–17. ISSN: 0272-1716.

[Gho+11] Abhijeet Ghosh et al. "Multiview face capture using polarized spherical gradient illumination". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 30.6 (2011), 129:1–129:10.

[Gol+06] A. Golovinskiy et al. "A Statistical Model for Synthesis of Detailed Facial Geometry". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 25.3 (2006), pp. 1025–1034.

[GP03] Martin A. Giese and Tomaso Poggio. "Neural mechanisms for the recognition of biological movements". In: *Nat Rev Neurosci* 4.3 (2003), pp. 179 –192.

[Gro+10] Max Grosse et al. "Coded aperture projection". In: *ACM Trans. Graph.* 29.3 (2010), 22:1–22:12. ISSN: 0730-0301.

[Gru13] Anselm Grundhöfer. "Practical Non-linear Photometric Projector Compensation". In: *2nd Int. Workshop on Computational Cameras and Display*. 2013.

[Gue+98] Brian Guenter et al. "Making Faces". In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '98. ACM, 1998, pp. 55–66. ISBN: 0-89791-999-8.

[Har+06] Michael Harville et al. "Practical methods for geometric and photometric correction of tiled projector". In: *Computer Vision and Pattern Recognition Workshop*. 2006, pp. 5–5.

[Hor87] Berthold K. P. Horn. "Closed-form solution of absolute orientation using unit quaternions". In: *J. Opt. Soc. Am. A* 4.4 (1987), pp. 629–642.

[HPL06] Parag Havaldar, F Pighin, and JP Lewis. "Performance driven facial animation". In: *ACM SIGGRAPH Courses*. 2006.

[Hua+11a] Haoda Huang et al. “Controllable hand deformation from sparse examples with rich details”. In: *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. 2011, pp. 73–82.

[Hua+11b] Haoda Huang et al. “Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition”. In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 30.4 (2011), 74:1–74:10.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[Ish06] Hiroshi Ishiguro. “Interactive humanoids and androids as ideal interfaces for humans”. In: *Proc. International Conference on Intelligent user interfaces*. Sydney, Australia, 2006, pp. 2–9. ISBN: 1-59593-287-9.

[JG10] Jorge Jimenez and Diego Gutierrez. “GPU Pro: Advanced Rendering Techniques”. In: ed. by Wolfgang Engel. AK Peters Ltd., 2010. Chap. Screen-Space Subsurface Scattering, pp. 335–351.

[Jon+06] A. Jones et al. “Performance Geometry Capture for Spatially Varying Relighting”. In: *3rd European Conference on Visual Media Production (CVMP 2006)*. 2006.

[K+02] Kolja Kähler et al. “Head Shop: Generating Animated Head Models with Anatomical Structure”. In: *Proc. SCA*. 2002, pp. 55–63.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. “Poisson surface reconstruction”. In: *Proc. SGP*. 2006.

[KH12] Martin Klaudiny and Adrian Hilton. “High-detail 3D capture and non-sequential alignment of facial performance”. In: *3DIMPVT*. 2012.

[Kim+13] Doyub Kim et al. “Near-exhaustive Precomputation of Secondary Cloth Effects”. In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 32.4 (2013), 87:1–87:8.

[KT03] S. Kshirsagar and N. M. Thalmann. “Visyllable based speech animation”. In: *Computer Graphics Forum (Proc. Eurographics)* 22.3 (2003).

[Kur+11] T. Kuratate et al. “Mask-bot: A life-size robot head using talking head animation for human-robot communication”. In: *Int. Conference on Humanoid Robots (Humanoids)*. 2011, pp. 99 –104.

[Law+11] Alvin J. Law et al. “Perceptually Based Appearance Modification for Compliant Appearance Editing.” In: *Comput. Graph. Forum* 30.8 (2011), pp. 2288–2300.

[LBB02] Sooha Park Lee, Jeremy B. Badler, and Norman I. Badler. “Eyes Alive”. In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 21.3 (2002), pp. 637–644.

[LC04] Caroline Larboulette and Marie-Paule Cani. "Real-Time Dynamic Wrinkles". In: *Proc. Computer Graphics Int.* 2004, pp. 522–525.

[LCF00] J.P. Lewis, M. Cordner, and N. Fong. "Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation". In: *Proc. SIGGRAPH 00*. 2000, pp. 165–172.

[Lew+05] J.P. Lewis et al. "Reducing blendshape interference by selected motion attenuation". In: *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*. 2005, pp. 25–29.

[Lew+14] JP Lewis et al. "Practice and theory of blendshape facial models". In: *EUROGRAPHICS STAR report* (2014), pp. 199–218.

[Li+13] Hao Li et al. "Realtime facial animation with on-the-fly correctives". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 32.4 (2013), 42:1–42:10.

[Li+15] Jun Li et al. "Lightweight wrinkle synthesis for 3D facial modeling and animation". In: *Computer-Aided Design* 58.0 (2015), pp. 117–122.

[Lin+09] Peter Lincoln et al. "Animatronic Shader Lamps Avatars". In: *Proc. Int. Symposium on Mixed and Augmented Reality*. 2009, pp. 27–33. ISBN: 978-1-4244-5390-0.

[Lip+05] Yaron Lipman et al. "Linear rotation-invariant coordinates for meshes". In: *ACM Trans. Graph.* 24.3 (2005), pp. 479–487.

[LMD12] B. H. Le, Xiaohan Ma, and Zhigang Deng. "Live Speech Driven Head-and-Eye Motion Generators". In: *IEEE TVCG* 18.11 (2012), pp. 1902–1914.

[LRF93] H. Li, P. Roivainen, and R. Forchheimer. "3-D motion estimation in model-based facial image coding". In: *IEEE PAMI* 15.6 (1993), pp. 545–555.

[LSP08] Hao Li, Robert W. Sumner, and Mark Pauly. "Global Correspondence Optimization for Non-Rigid Registration of Depth Scans". In: *Computer Graphics Forum (Proc. SGP)* 27.5 (2008).

[LWP10] Hao Li, Thibaut Weise, and Mark Pauly. "Example-based facial rigging". In: *ACM Trans. Graph.* 29.4 (2010), p. 32.

[Ma+04] J. Ma et al. "Accurate automatic visible speech synthesis of arbitrary 3D model based on concatenation of diviseme motion capture data". In: *Computer Animation and Virtual Worlds* 15 (2004), pp. 1–17.

[Ma+07] W.-C. Ma et al. "Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination". In: *Eurographics Symposium on Rendering*. 2007, pp. 183–194.

[Ma+08] Wan-Chun Ma et al. "Facial Performance Synthesis using Deformation-Driven Polynomial Displacement Maps". In: *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 27.5 (2008), 121:1–121:10.

[MB13] Anush K Moorthy and Alan C Bovik. "A survey on 3D quality of experience and 3D quality assessment". In: *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics. 2013, pp. 86510M–86510M.

[MC10] Matthias Müller and Nuttapong Chentanez. "Wrinkle Meshes". In: *Proc. SCA*. 2010, pp. 85–92.

[MEB12] Samer Al Moubayed, Jens Edlund, and Jonas Beskow. "Taming Mona Lisa: Communicating gaze faithfully in 2D and 3D facial projections". In: *ACM Trans. Interact. Intell. Syst.* 1.2 (2012). ISSN: 2160-6455.

[MI06] Karl F. MacDorman and Hiroshi Ishiguro. "The uncanny advantage of using androids in cognitive and social science research". In: *Interaction Studies* 7.3 (2006), pp. 297–337.

[MIR12] Kana Misawa, Yoshio Ishiguro, and Jun Rekimoto. "Ma petite cherie: what are you looking at?: a small telepresence system to support remote collaborative work for intimate communication". In: *Proc. Augmented Human International Conference*. AH '12. Meg
UTF00E8ve, France: ACM, 2012, 17:1–17:5. ISBN: 978-1-4503-1077-2.

[MLD09] Xiaohan Ma, Binh Huy Le, and Zhigang Deng. "Style learning and transferring for facial animation editing". In: *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New Orleans, Louisiana, 2009, pp. 123–132. ISBN: 978-1-60558-610-6.

[MMK12] M. Mori, K.F. MacDorman, and N. Kageki. "The Uncanny Valley [From the Field]". In: *Robotics Automation Magazine, IEEE* 19.2 (2012), pp. 98–100. ISSN: 1070-9932.

[Mor70] Masahiro Mori. "The Uncanny Valley". In: *Energy 7*. Vol. 4. 1970, pp. 33–35.

[MT+02] N. Magnenat-Thalmann et al. "A Computational Skin Model: Fold and Wrinkle Formation". In: *Trans. Info. Tech. Biomed.* 6.4 (2002), pp. 317–323.

[NIH07a] S. Nishio, H. Ishiguro, and N. Hagita. "Humanoid Robots: New Developments". In: ed. by Amando Carlos de Pina Filho. I-Tech, 2007. Chap. Geminoid: Teleoperated Android of an Existing Person.

[NIH07b] Shuichi Nishio, Hiroshi Ishiguro, and Norihiro Hagita. *Geminoid: Teleoperated android of an existing person*. INTECH Open Access Publisher, 2007.

[NIS11] Momoyo Nagase, Daisuke Iwai, and Kosuke Sato. "Dynamic defocus and occlusion compensation of projected imagery by model-based optimal projector selection in multi-projection environment". In: *Virtual Real.* 15.2-3 (2011), pp. 119–132. ISSN: 1359-4338.

[NN01] Jun-yong Noh and Ulrich Neumann. "Expression Cloning". In: *Proc. SIGGRAPH 01*. 2001, pp. 277–288.

[OS08] Yuji Oyamada and Hideo Saito. "Defocus Blur Correcting Projector-Camera System". In: *Proc. Int. Conference on Advanced Concepts for Intelligent Vision Systems*. Juan-les-Pins, France, 2008, pp. 453–464. ISBN: 978-3-540-88457-6.

[Ota+12] Miguel A. Otaduy et al. "Data-driven Simulation Methods in Computer Graphics: Cloth, Tissue and Faces". In: *ACM SIGGRAPH 2012 Courses*. SIGGRAPH '12. Los Angeles, California: ACM, 2012, 12:1–12:96. ISBN: 978-1-4503-1678-1.

[Par74] F. I. Parke. "A parametric model for human faces". PhD thesis. University of Utah, 1974.

[PE12] Matthew F Peterson and Miguel P Eckstein. "Looking just below the eyes is optimal across face recognition tasks". In: *Proceedings of the National Academy of Sciences* 109.48 (2012).

[PM90] P. Perona and J. Malik. "Scale-space and edge detection using anisotropic diffusion". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12.7 (1990), pp. 629–639. ISSN: 0162-8828.

[Pop+09] Tiberiu Popa et al. "Wrinkling Captured Garments Using Space-Time Data-Driven Deformation". In: *Computer Graphics Forum (Proc. Eurographics)* 28.2 (2009), pp. 427–435.

[PSS99] Frederic H. Pighin, Richard Szeliski, and David Salesin. "Resynthesizing Facial Animation through 3D Model-based Tracking". In: *ICCV*. 1999, pp. 143–150.

[Pyu+03] H. Pyun et al. "An example-based approach for facial expression cloning". In: *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2003, pp. 167–176.

[Ras+01] Ramesh Raskar et al. "Shader Lamps: Animating Real Objects With Image-Based Illumination". In: *Proc. Eurographics Workshop on Rendering Techniques*. 2001, pp. 89–102. ISBN: 3-211-83709-4.

[Ras+98] Ramesh Raskar et al. "The office of the future: A unified approach to image-based modeling and spatially immersive displays". In: *Proc. Conf. on Comp. Graph. and Int. Techniques*. 1998, pp. 179–188.

[Rhe+11] Taehyun Rhee et al. "Real-time Facial Animation from Live Video Tracking". In: *Proc. SCA*. 2011.

[Roh+10] Damien Rohmer et al. "Animation Wrinkling: Augmenting Coarse Cloth Simulations with Realistic-looking Wrinkles". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 29.6 (2010), 157:1–157:8.

[Ruh+14] K Ruhland et al. "Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems". In: *Eurographics State of the Art Reports*. 2014, pp. 69–91.

[SA07] Olga Sorkine and Marc Alexa. "As-rigid-as-possible surface modeling". In: *Symposium on Geometry processing*. Vol. 4. 2007.

[Sag+94] Mark A. Sagar et al. "A virtual environment and model of the eye for surgical simulation". In: *Proceedings of Computer Graphics and Interactive Techniques*. 1994, pp. 205–212.

[Say+11] Ayse Pinar Saygin et al. "The thing that should not be: predictive coding and the uncanny valley in perceiving human and humanoid robot actions". In: *Social Cognitive and Affective Neuroscience* (2011).

[Sei+06] Steven M Seitz et al. "A comparison and evaluation of multi-view stereo reconstruction algorithms". In: *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2006, pp. 519–528.

[Sen+05] Pradeep Sen et al. "Dual photography". In: *ACM Trans. Graph.* 24.3 (2005), pp. 745–755.

[Seo+11] Yeongho Seol et al. "Artist Friendly Facial Animation Retargeting". In: *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 30.6 (2011), 162:1–162:10.

[Seo+12] Yeongho Seol et al. "Spacetime expression cloning for blendshapes". In: *ACM Trans. Graph.* 31.2 (2012), p. 14.

[Shi+14] Fuhao Shi et al. "Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 33 (6 2014).

[Sif+06] Eftychios Sifakis et al. "Simulating Speech with a Physics-based Facial Muscle Model". In: *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '06. Vienna, Austria: Eurographics Association, 2006, pp. 261–270. ISBN: 3-905673-34-7.

[SJ13] Mohamed Sathik and Sofia G. Jonathan. "Effect of facial expressions on student's comprehension recognition in virtual educational environments". In: *Springerplus* 2 (2013). 558[PII], p. 455. ISSN: 2193-1801.

[SKSS14] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. "Total Moving Face Reconstruction". In: *ECCV*. 2014.

[Smi+05] Marie L Smith et al. "Transmitting and decoding facial expressions". In: *Psychological Science* 16.3 (2005), pp. 184–189.

[SMW06] Scott Schaefer, Travis McPhail, and Joe Warren. "Image deformation using moving least squares". In: *ACM Trans. Graph.* Vol. 25. 3. 2006, pp. 533–540.

[SN07] Jun'ichiro Seyama and Ruth S. Nagayama. "The Uncanny Valley: Effect of Realism on the Impression of Artificial Human Faces". In: *Presence: Teleoper. Virtual Environ.* 16.4 (2007), pp. 337–351. ISSN: 1054-7460.

[SNF05] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. "Automatic determination of facial muscle activations from sparse motion capture marker data". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 24.3 (2005), pp. 417–425.

[Sor+04] Olga Sorkine et al. "Laplacian Surface Editing". In: *Proc. SGP*. Nice, France: ACM Press, 2004, pp. 179–188.

[SP04] B. Sumner and J. Popović. "Deformation transfer for triangle meshes". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 23.3 (2004), pp. 399–405.

[SSH12] M. Seiler, J. Spillmann, and M. Harders. "Enriching Coarse Interactive Elastic Objects with High-resolution Data-driven Deformations". In: *Proc. SCA*. 2012, pp. 9–17.

[SSM01] Rahul Sukthankar, Robert G Stockton, and Matthew D Mullin. "Smarter presentations: Exploiting homography in camera-projector systems". In: *Proc. Int. Conference on Computer Vision*. Vol. 1. IEEE. 2001, pp. 247–253.

[Tak+11] Kenshi Takayama et al. "GeoBrush: Interactive Mesh Geometry Cloning". In: *Computer Graphics Forum* 30.2 (2011), pp. 613–622.

[Ten+06] J Rafael Tena et al. "A validated method for dense non-rigid 3D face registration". In: *Int. Conf. on Video and Signal Based Surveillance*. 2006.

[Tru+11] Laura C. Trutoiu et al. "Modeling and Animating Eye Blinks". In: *ACM Trans. Appl. Percept.* 8.3 (2011).

[TTM11] J. Rafael Tena, Fernando De la Torre, and Iain Matthews. "Interactive Region-Based Linear 3D Face Models". In: *ACM Trans. Graph. (Proc. ACM SIGGRAPH)* 30.4 (2011), 76:1–76:10.

[TW93] D. Terzopoulus and K. Waters. "Analysis and synthesis of facial image sequences using physical and anatomical models". In: *IEEE Trans. PAMI* 14 (1993), pp. 569–579.

[Val+12] L. Valgaerts et al. "Lightweight Binocular Facial Performance Capture under Uncontrolled Lighting". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 31.6 (2012).

[Vla+05] Daniel Vlasic et al. "Face transfer with multilinear models". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 24.3 (2005), pp. 426–433. ISSN: 0730-0301.

[VLR05] Kartik Venkataraman, Suresh Lodha, and Raghu Raghavan. "A kinematic-variational model for animating skin with wrinkles." In: *Computers & Graphics* 29.5 (2005), pp. 756–770.

[Wan+04a] Yang Wang et al. "High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions". In: *Computer Graphics Forum* 23.3 (2004), pp. 677–686.

[Wan+04b] Zhou Wang et al. "Image quality assessment: From error visibility to structural similarity". In: *Trans. on Image Processing* 13.4 (2004), pp. 600–612.

[Wan+10] Huamin Wang et al. "Example-based Wrinkle Synthesis for Clothing Animation". In: *ACM Trans. Graphics (Proc. SIGGRAPH)* 29.4 (2010), 107:1–107:8.

[Wat87] Keith Waters. "A Muscle Model for Animating Three-Dimensional Facial Expression". In: *Computer Graphics (Proc. SIGGRAPH)*. 1987, pp. 17–24.

[WB07] G. Wetzstein and O. Bimber. "Radiometric Compensation through Inverse Light Transport". In: *Computer Graphics and Applications, 2007. PG '07.* 2007, pp. 391 –399.

[Wei+11] Thibaut Weise et al. "Realtime Performance-Based Facial Animation". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 30.4 (2011), 77:1–77:10.

[Wen+05] A. Wenger et al. "Performance Relighting and Reflectance Transformation with Time-Multiplexed Illumination". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 24.3 (2005), pp. 756–764.

[Wey+06] Tim Weyrich et al. "Analysis of human faces using a measurement-based skin reflectance model". In: *ACM Trans. Graph.* 25.3 (2006), pp. 1013–1024.

[Wil+10] C.A. Wilson et al. "Temporal Upsampling of Performance Geometry Using Photometric Alignment". In: *Trans. Graph.* 29.2 (2010).

[Wil90] Lance Williams. "Performance-Driven Facial Animation". In: *Computer Graphics (Proc. SIGGRAPH)*. 1990, pp. 235–242.

[WKMT96] Yin Wu, Prem Kalra, and Nadia Magnenat-Thalmann. "Simulation of Static and Dynamic Wrinkles of Skin". In: *Proc. Computer Animation*. 1996, pp. 90–97.

[WLO10] Axel Weissenfeld, Kang Liu, and Jörn Ostermann. "Video-realistic image-based eye animation via statistically driven state machines". In: *The Visual Computer* 26.9 (2010), pp. 1201–1216.

[WM14] Mark Warburton and Steve Maddock. "Physically-based forehead animation including wrinkles". In: *Comput. Animat. Virtual Worlds* (2014).

[Wu+11] Chenglei Wu et al. "Shading-based Dynamic Shape Refinement from Multi-view Video under General Illumination". In: *ICCV*. 2011.

[ZBO13] J. S. Zurdo, J. P. Brito, and M. A. Otaduy. "Animating Wrinkles by Example on Non-Skinned Cloth". In: *IEEE TVCG* 19.1 (2013), pp. 149–158.

[ZH06] S. Zhang and P. Huang. "High-resolution, real-time three-dimensional shape measurement". In: *Optical Engineering* 45.12 (2006), p. 123601.

[Zha+04] L. Zhang et al. "Spacetime Faces: High Resolution Capture for Modeling and Animation". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 23.3 (2004), pp. 548–558.

[Zha00] Zhengyou Zhang. "A Flexible New Technique for Camera Calibration". In: *Trans. Pattern Anal. Mach. Intell.* 22.11 (2000), pp. 1330–1334. ISSN: 0162-8828.

[ZN06] Li Zhang and Shree Nayar. "Projection defocus analysis for scene capture and image display". In: *ACM Trans. Graph.* 25.3 (2006), pp. 907–915. ISSN: 0730-0301.

[ZS05] Y. Zhang and T. Sim. "Realistic and efficient wrinkle simulation using an anatomy-based face model with adaptive refinement". In: *Computer Graphics International 2005*. 2005, pp. 3–10.

[ZST05] Y. Zhang, T. Sim, and C.L. Tan. "Simulating wrinkles in facial expressions on an anatomy-based face". In: *Proc. ICCS*. 2005, pp. 207–215.