Diss. ETH No. 20672

Passive Spatio-Temporal Geometry Reconstruction of Human Faces at Very High Fidelity

A dissertation submitted to
ETH Zurich

for the Degree of
Doctor of Sciences

presented by
Thabo Beeler
MSc ETH CS, Switzerland
born Nov. 2, 1978
citizen of Flums-Kleinberg, Switzerland

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Prof. Dr. Thomas Vetter, co-examiner
Dr. Paul Beardsley, co-examiner

2012

Abstract

The creation of realistic synthetic human faces is one of the most important and at the same time also most challenging topics in computer graphics. The high complexity of the face as well as our familiarity with it renders manual creation and animation impractical. The method of choice is thus to capture both shape and motion of the human face from the real life talent. To date, this is accomplished using active techniques which either augment the face either with markers or project specific illumination patterns onto it. Active techniques currently provide the highest geometric accuracy, but they have severe shortcomings when it comes to capturing performances.

In this thesis we present an entirely passive and markerless system to capture and reconstruct facial performances at un-preceded spatio-temporal resolution. The proposed algorithms compute the facial shape and motion at skin-pore resolution from multiple cameras producing per frame temporally compatible geometry.

The thesis contains several contributions, both in computer vision and computer graphics. We introduce multiple capture setups that employ off-the-shelf cameras and are tailored to capturing the human face. We also propose different illumination setups, including the design and construction of a multi-purpose light stage, with capabilities that reach beyond of what is required within this thesis. The light stage contains $\sim$500 color LEDs that can be controlled individually to produce arbitrary spatio-temporal illumination patterns. We present a practical calibration technique designed to automatically calibrate face capture setups as well as techniques to geometrically calibrate the light stage.

The core contribution of this thesis is a novel multi-view stereo (MVS) algorithm that introduces the concept of *Mesoscopic Augmentation*. We demonstrate that this algorithm can reconstruct facial skin at quality on-par with active techniques. The system is single shot in that it requires only a single exposure per camera to reconstruct the facial geometry, which enables it to reconstruct even ephemeral poses and makes it well suited for performance capture. We extend the proposed MVS algorithm by the concept of the *Episurface*, which provides a plausible approximation to the true skin surface in areas where it is occluded by facial hair. We also present the first algorithm to reconstruct sparse facial hair at hair fiber resolution from a single exposure.

To track skin movement over time without the use of markers we propose an algorithm that employs optical flow. To overcome inherent limitations of

optical flow, such as drift, we introduce the concept of *Anchor Frames*, which enables us to track facial performances robustly even over long periods of time. Most optical flow algorithms assume some sort of brightness constancy. This assumption, however, is violated for deforming surfaces, as the deformation changes self-shading over time. We present a technique called *Ambient Occlusion Cancelling*, which leverages the reconstructed per-frame geometry to remove varying self-shading from the images. We demonstrate that this technique complements and substantially improves existing optical flow methods. In addition, we show how the varying self-shading can be used to improve the reconstructed geometry.

The concepts and ideas presented in this thesis have the potential to inspire future research in the area of time-varying geometry reconstruction. Already, several concepts presented in thesis have been used in industry to help produce the next generation CG faces in theme parks, computer games, and feature films.

Zusammenfassung

Das Nachbilden von realistischen menschlichen Gesichtern ist eines der wichtigsten und gleichzeitig herausforderndsten Gebiete der Computer Grafik. Menschliche Gesichter manuell zu erstellen und zu animieren ist aufgrund der hohen Komplexität und unserem starken Bezug zum menschlichen Gesicht impraktikabel. Die Alternative ist daher sowohl die Form als auch die Bewegung des Gesichtes direkt vom Darsteller zu übernehmen und zu digitalisieren. Bis anhin geschieht dies mittels aktiver Techniken, welche das Gesicht einerseits physisch durch das Anbringen von Markierungen, andererseits optisch durch die Projektion von Mustern, verändern. Aktive Techniken liefern gegenwärtig die höchste geometrische Genauigkeit, haben jedoch einige Nachteile bei der Aufnahme von zeitlich variierender Mimik — wie z.B. niedrige zeitlich-räumliche Auflösung, teure Hardware oder Verfälschung der Textur.

In dieser Dissertation präsentieren wir ein vollkommen passives und markierungsfreies System, um zeitlich variierende Gesichtsmimik digital zu rekonstruieren. Sowohl die Form wie auch die Bewegung des Gesichtes wird von mehreren Kameras erfasst und auf Poren-Ebene rekonstruiert. Das Endprodukt ist zeitlich kompatible dreidimensionale Geometrie mit bis anhin unerreicht hoher zeitlich-räumlicher Auflösung.

Die Dissertation leistet mehrere Beiträge, sowohl im Gebiet Computer Vision wie auch in der Computer Grafik. Wir präsentieren mehrere Aufbauten, die geeignet sind das menschliche Gesicht zu erfassen. Da die Aufbauten handelsübliche Kameras einsetzen, sind sie einfach und kosteneffizient replizierbar. Zusätzlich stellen wir verschiedene Beleuchtungsmethoden vor, vom handelsüblichen, polarisierten Kamerablitz bis hin zu unserer Lichtbühne. Die Lichtbühne wurde, mit Blick auf zukünftigen Anwendungen, möglichst flexibel konzipiert. Sie besteht aus $\sim$500 farbigen LEDs, welche individuell angesteuert werden können, um arbiträre zeitlich-räumliche Beleuchtungskonditionen zu produzieren. Wir präsentieren praxisorientierte Methoden, um die vorgestellten Kameraaufbauten und die Lightstage automatisch zu kalibrieren.

Das Kernstück der Arbeit ist ein neuer multi-view stereo Algorithmus (MVS), welcher das Konzept der *Mesoskopischen Augmentierung* einführt. Wir demonstrieren, dass dieser Algorithmus menschliche Gesichter mit einer Qualität rekonstruieren kann, welche jener von aktiven Methoden nicht

nachsteht. Das System ist 'single-shot', da es nur ein einzelnes Bild pro Kamera benötigt, um das Gesicht zu rekonstruieren — wodurch selbst flüchtige Momente eingefangen werden können. Wir erweitern den MVS Algorithmus durch das Konzept der *Episurface*, welche eine plausible Oberfläche bietet wo die Haut von Haarwuchs bedeckt ist. Zudem stellen wir den ersten single-shot Algorithmus vor, der Haare einzeln zu rekonstruieren vermag.

Des Weiteren stellen wir einen Algorithmus vor, der mithilfe von 'Optical Flow' die Bewegung der Haut direkt von der Textur berechnet und somit keine Markierungen benötigt. Um die inhärenten Limitationen von 'Optical Flow', wie z.B. Drift, zu umgehen, stellen wir das Konzept der *Anchor Frames* vor. Dadurch wird es möglich, zeitlich variierende Mimik robust selbst über lange Zeit hinweg zu rekonstruieren. Die meisten 'Optical Flow' Algorithmen gehen davon aus, dass sich die Intensität eines Punktes über die Zeit hinweg nicht verändert. Diese Annahme ist aber bei sich verformenden Oberflächen nicht gegeben, da die sich verändernde Form die Schattierung der Oberfläche beeinflusst. Wir stellen eine Methode vor, *Cancelling Ambient Occlusion*, die mithilfe der errechneten Geometrie sich verändernde Schattierung aus den Bildern entfernt. Diese Methode komplementiert bestehende 'Optical Flow' Algorithmen und wir zeigen, dass ihre Genauigkeit dadurch substantiell verbessert wird. Zudem zeigen wir, wie die variierende Schattierung verwendet werden kann, um die errechnete Geometrie zu verbessern.

Wir hoffen, dass die vorgestellten Konzepte und Ideen künftiger Forschung auf dem Gebiet der Erfassung und Rekonstruktion zeitlich variierender Geometrie als Inspiration dienen. Bereits jetzt tragen verschiedene der vorgestellten Konzepte dazu bei die nächste Generation von CG Gesichtern für Vergnügungsparks, Computer Spiele und Filme zu erschaffen.

For Maya

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Dr. Markus Gross. He enabled, encouraged and supported me during the course of my thesis. His apparent never ending interest in open research problems and his enthusiasm to push the envelope of the state-of-the-art were a great inspiration.

Second, I would like to thank my close collaborators; Bernd Bickel who talked me into the topic of face capture and continuously supported me, Paul Beardsley and Bob Sumner who shared the burden of supervising me as well as Derek Bradley with his unconditional commitment to and profound knowledge of face capturing. Their experience, technical knowledge and strategic guidance formed the fundamental basis of my Ph.D.

Further, it was a great pleasure to work with Fabian Hahn, Henning Zimmer, Gioacchino Noris, Steve Marschner and Craig Cotsman. It was an invaluable experience to consult, discuss and work with them, and my thesis would not be nearly what it is without their support.

Thanks a lot to Paul Beardsley, Bernd Bickel, Derek Bradley, Maya Sigron and Bob Sumner for proof reading parts of my thesis.

I owe all people of Disney Research Zurich and the Computer Graphics Laboratory a debt of gratitude. They formed the professional and social environment that inspired and motivated me every day anew; Prateek Agarwal, Tunç Aydin, Jeroen van Baar, Miklos Balint, Ilya Baran, Gian-Marco Baschera, Katie Basset, Paul Beardsley, Amit Bermano, Bernd Bickel, Derek Bradley, Alexandre Chapiro, Stelian Coros, Ronnie Gaensli, Marcel Germann, Domenico Giustiniano, Pierre Greisen, Max Grosse, Anselm Grundhofer, Ralf Habel, Simon Heinzle, Alexander Hornung, Wojciech Jarosz, Tanja Käser, Peter Kaufmann, Changil Kim, Claudia Kuster, Manuel Lang, Stefan Mangold, Sebastian Martin, Javier Alonso Mora, Maurizio Nitti, Derek Nowrouzezahrai, Thomas Oskam, Cengiz Oztirelli, Marios Papas, Tiberiu Popa, Steven Poulakos, Yael Pritsch, Nico Ranieri, Christian Regg, Rafael Monroy Rodrigues, Gerhard Röthlin, Johannes Schmid, Mélina Skouras, Aljoscha Smolic, Barbara Solenthaler, Nikolce Stephanoski, Bob Sumner, Bernhard Thomaszewski, Vladimir Vukadinovic, Oliver Wang, and Fabio Zuend.

Special thanks to Hannes Diethelm, Ronnie Gaensli and Timon Meier — their technical skill made the design and construction of our light stage possible.

Very special thanks to the strong lab admin team; Michelle Berchtold, Linda Breu, Sarah Disch, Markus Portmann, Lioudmila Thalmann and Stephan Veen. They helped me facing everyday challenges countless times. Very special thanks also to Peter Kaufmann for letting me use his great development framework, and to Maurizio Nitti for helping me out with his amazing artistic skills. Many thanks to my lab mate Gioacchino Noris. It was great fun to share the office with him and he deserves a medal for coping with me and my chaotic nature.

Furthermore, I want to thank all of our patient actors. Special thanks goes to the 'star actors'; Leila Gangji, Manuel Lang and Sean James Sutton.

Finally, I want to thank all my friends, my family and my girlfriend for all their support and understanding. They formed my backbone during these three work intensive years.

Contents

CHAPTER 1

Introduction

Creating realistic human faces is amongst the biggest challenges in computer graphics. At the same time, it is also one of the most important tasks and of great interest for many different domains — for surgery previsualization and planning in medical fields, humanoid robotics, virtual reality, synthetic storytelling, archival purposes and special effects in the movie and game industries, where it is sometimes called the *Holy Grail of special effects*.

A core component of the challenge in creating synthetic human faces that are indistinguishable from real faces comes from the high complexity of the human face. The skin of a human face exhibits very detailed geometric structure, intricate appearance variation due to underlying tissues and highly non-linear deformation properties. These properties vary both spatially and temporally as they are influenced by physiological and anatomical effects such as blood flow or the underlying bone structure. In addition, the human face has other features, such as eyes or facial hair, that exhibit very different properties and contribute substantially to the overall appearance.

The difficult task of building realistic faces is aggravated by human perception. Recognizing and interpreting faces is a fundamental part of human perception, refined through evolution. This ability has been essential for survival as it enabled to predict behavior, spot potential aggression and identify sick individuals. We can effortlessly discriminate even very subtle expres-

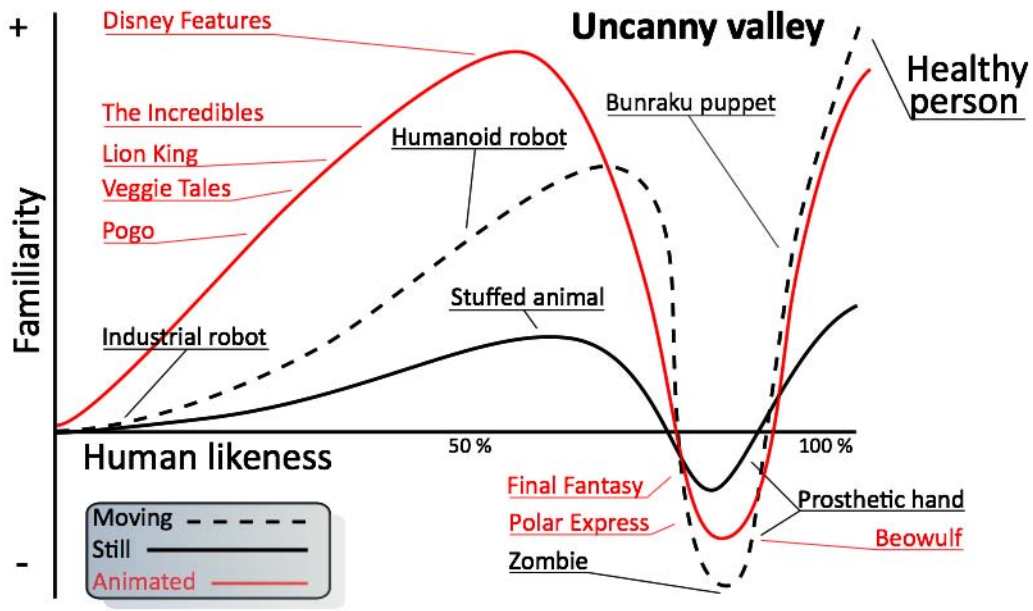


Figure 1.1: Uncanny valley - *The graph describes the observation that characters which are similar to humans, but not perfectly human-like, provoke a negative emotional response. The hypothesis is that human perception cannot easily decide whether a stimuli is real or synthetic in this area, thus requiring additional and longer cognitive processing which in turn leads to an adverse response [Cheetham et al., 2011]. The hypothesis was originally introduced by Mori in the 1970s [Mori, 1970] in the field of robotics and later extended to animated films, where a similar effect can be observed [Joly, 2012].*

sion changes, such as a twitch of an eye, which might completely change the perception of an expression (e.g., fake vs. sincere smile). These effects may only last fractions of a second but are important aspects of human communication. Failure to reproduce these subtle spatio-temporal effects does not simply reduce realism but can provoke a psychological reaction called the *uncanny valley*. As depicted in Figure 1.1, classification of faces that are neither fully realistic nor sufficiently abstract poses a challenge to human perception. As a consequence, these faces disturb the viewer who thus refuses to accept them.

To overcome the uncanny valley, a synthetic face has to reproduce both *accurate dynamics* and *detailed geometry*. Accurate manual modeling and animation of these small-scale effects is tedious and hardly practical. Simulating them would be a much better alternative. However, the complex structure

and dynamics of a human face render realistic physical simulation unpractical to date. A third option is thus to capture and reconstruct real human faces, which has the potential to accurately reproduce all these subtle effects. To do so, a capture method with *very high spatio-temporal resolution* is required. In this thesis we present methods to reconstruct the facial geometry at skin pore level and at full framerate, achieving the **highest spatio-temporal resolution to date**.

1.1 Contributions

The primary **theoretical contributions** of the work are:

- A method for passive, single-shot, multi-view stereo reconstruction of a human face. This method provides high-quality reconstructions which had previously only been obtainable using active illumination.
- The concept of *Mesoscopic Augmentation*, which augments multi-view stereo geometry with shape-from-shading to provide an estimate of skin geometry at wrinkle and skin pore level.
- A method for coupled reconstruction of skin surface and sparse hair. We introduce the concept of an *Episurface* which is a putative underlying surface that is consistent with the observed facial hair. Hair is reconstructed fiber by fiber from a single exposure per camera.
- The concept of *Anchor Frames*, which enables robust passive markerless performance capture. This marks one of the first attempts to combat drift in tracking facial features over long image sequences.
- A technique called *Ambient Occlusion Cancelling* for removing self-shadowing effects from the captured images using the reconstructed geometry. We demonstrate that this approach complements and improves both existing optical flow methods and geometry reconstruction techniques. The concept also applies to other deforming objects, such as cloth.

The primary **system contributions** of the work are:

- An end-to-end pipeline to capture, reconstruct and track facial performances of human actors.
- A quantitative evaluation of the accuracy of the reconstruction technique by comparing it to laser-scanned ground truth data, demonstrat-

ing that passive MVS is now on a par with active reconstruction techniques.

- A quantitative evaluation of the robustness of the system by successfully reconstructing several thousand people of different ages, gender, ethnicities and under varying expressions without a single failure case.
- Several different camera setups, ranging from a single handheld stereo camera, over static and dynamic multi-camera systems to setups that combine different cameras and lenses to capture the face with spatially varying resolution.
- A practical and convenient calibration method designed to calibrate camera setups used for facial capture. Calibration is fully automatic from a single image per camera.
- A versatile and flexible multi-purpose light stage. The light stage is a spherical construct (2 meters in diameter) equipped with several hundred LEDs that can be controlled individually to produce arbitrary spatio-temporal illumination conditions. We also propose methods to calibrate the light stage geometrically.

1.2 Organization

The thesis is organized in three main parts. In the first part we present different capture setups and calibration techniques. The second part introduces methods to reconstruct static geometry of skin surface and facial hair. In the third and last part, static skin reconstruction is extended to include face dynamics.

In particular,

- **Chapter 2** presents a short introduction to the fundamentals and related work relevant to this thesis.
- **Chapter 3** discusses various capture setups that were designed and deployed in the course of the thesis. It addresses camera synchronization and introduces different illumination scenarios as well as *Helios*, our multi-purpose light stage.
- **Chapter 4** is devoted to calibration. *Multi Camera Calibration from a Spherical Object* is introduced as a convenient and accurate camera calibration technique for the setups proposed in Chapter 3. We further

propose algorithms to geometrically calibrate the multi-purpose light stage introduced in Chapter 3.

- **Chapter 5** presents the multi-view stereo (MVS) reconstruction pipeline as well as the concept of *Mesoscopic Augmentation*, which is the core geometry reconstruction technology all subsequent chapters are based on. It provides a thorough analysis of reconstruction accuracy and robustness, both quantitatively and qualitatively, and demonstrates that passive MVS systems are now able to compete with active techniques in terms of reconstruction quality for human faces.
- **Chapter 6** introduces methods to identify, remove and reconstruct sparse facial hair. The technology from Chapter 5 is extended with the concept of the *Episurface* — a surface that corresponds to the true skin surface wherever it is visible and provides a plausible substrate where it is covered by hair. We present a method to reconstruct skin Episurface and facial hair in a coupled fashion. Hair is reconstructed fiber by fiber from a single exposure per camera.
- **Chapter 7** extends the static reconstruction technique from Chapter 5 to the temporal domain. It introduces the concept of *Anchor Frames*, which is based on the observation that certain facial configurations reoccur on a regular basis during a performance. These reoccurring facial expressions are identified and used to overcome common problems of drift and occlusion in the tracking of facial features over an extended image sequence. We demonstrate the power of the concept by implementing a passive, markerless performance capture system.
- **Chapter 8** introduces *Ambient Occlusion Cancelling*, a method that complements and improves existing reconstruction and tracking techniques. It is based on the observation that deforming objects cause self-shadowing, for example when forming wrinkles, which has an adverse effect on spatio-temporal reconstruction. We present a method to approximate and cancel self-shadowing from the input images and demonstrate that this technique improves the performance of a wide spectrum of well-known optical flow techniques. Furthermore, the temporally varying shading is used to improve the reconstructed geometry.
- **Chapter 9** summarizes the thesis and main contributions, and suggests areas of potential future research.

1.3 Publications

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

T. BEELER, B. BICKEL, R. SUMNER, P. BEARDSLEY, M. GROSS.
High-Quality Single-Shot Capture of Facial Geometry.
In *Proceedings of ACM SIGGRAPH (Los Angeles, USA, July 25-29, 2010), ACM Transactions on Graphics, vol. 29, no. 3.*

T. BEELER, F. HAHN, D. BRADLEY, B. BICKEL, P. BEARDSLEY, C. GOTSMAN, R. SUMNER, M. GROSS.
High-quality passive facial performance capture using anchor frames.
In *Proceedings of ACM SIGGRAPH (Vancouver, Canada, Aug. 7-11, 2011), ACM Transactions on Graphics, vol. 30, no. 4.*

T. BEELER, B. BICKEL, G. NORIS, S. MARSCHNER, P. BEARDSLEY, R. SUMNER, M. GROSS.
Coupled 3D Reconstruction of Sparse Facial Hair and Skin.
In *Proceedings of ACM SIGGRAPH (Los Angeles, USA, Aug. 5-9, 2012), ACM Transactions on Graphics, vol. 31, no. 3.*

T. BEELER, D. BRADLEY, Z. HENNING, M. GROSS.
Improved Reconstruction of Deforming Surfaces by Cancelling Ambient Occlusion.
In *European Conference on Computer Vision (ECCV) (Firenze, Italy, Oct 7-13, 2012).*

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

B. BICKEL, P. KAUFMANN, M. SKOURAS, B. THOMASZEWSKI, T. BEELER, D. BRADLEY, P. JACKSON, S. MARSCHNER, W. MATUSIK, M. GROSS.
Physical Face Cloning
In *Proceedings of ACM SIGGRAPH (Los Angeles, USA, Aug. 5-9, 2012), ACM Transactions on Graphics, vol. 31, no. 3.*

Additional publications written during the time period of this thesis include:

T. BEELER, B. BICKEL, R. SUMNER, P. BEARDSLEY, M. GROSS.
High-Quality Single-Shot Capture of Facial Geometry: Implementation Details.
Technical Report No. 671, Institute of Visual Computing, ETH Zuerich, 2010.

CHAPTER 2

Related Work

In this chapter we will review the related work for this thesis. In Section 2.1, we give an overview of surface reconstruction techniques. We limit our discussion to techniques that are applicable to face capture. In Section 2.2, we review related work on hair acquisition and reconstruction. In Section 2.3, we give an overview of existing performance capture approaches. Section 2.4 discusses optical flow, in particular illumination-robust approaches, which is an integral component for many tracking systems.

2.1 Surface Capture and Reconstruction

Surface reconstruction has a huge variety of applications, ranging from large-scale landscape and urban reconstructions over heritage scanning all the way to medical appliances. The available reconstruction techniques are just as diverse as the applications themselves and we will therefore limit ourselves to work related to capturing and reconstructing human faces. The available techniques are either active or passive in nature and they either estimate the position of the surface or its relative shape through normal estimation, or combine both position and normal estimation.

Active vs. Passive Active capture devices consist both of sensing and emitting parts, while passive capture devices only contain sensing components. Active capture devices can thus augment the scene with additional information such as structured light to facilitate reconstruction. On the downside, active techniques can often only be combined through temporal multiplexing, as they would interfere with each-other. Temporal multiplexing however is problematic when capturing dynamic content. Furthermore, active techniques are often limited in viewpoint and have problems with self occlusions. Passive capture devices have to rely solely on content that is already present in the scene, limiting their application range as compared to active techniques. They have typically less complex illumination requirements and are less distractive for the person being captured.

Position vs. Normal Estimation We identified two different concepts for reconstructing surfaces while reviewing related work. Position estimation techniques (Section 2.1.1) directly estimate the position of the surface in space. Techniques that belong to this class are usually robust and well suited to recover coarse geometry. However, they fail to recover small-scale surface variation. Normal estimation techniques (Section 2.1.2) on the other hand estimate the orientation of the surface given by its normal field. As the normal field is very sensitive to small variation of the surface, these techniques excel at recovering fine-scale structure. The shape of the surface can be recovered through integration of the normal field. In practice, integration often introduces low-frequency bias and noise, making these techniques less suited to reconstruct coarse geometry. This complementary nature has led to several systems that combine the two techniques, some of which are addressed in Section 2.1.3.

2.1.1 Position Estimation

The techniques presented in this section directly estimate the 3D position of the surface. Both active (e.g., *laser scanning, time-of-flight, structured light*) and passive (e.g., *passive stereo*) techniques exist.

Laser Scanning Laser scanners rasterize the surface by sweeping a point or line over it and compute the depth of the surface at these points using triangulation [Blais et al., 2004, Saint-Marc et al., 1991]. Laser scanners provide very dense and accurate measurements for static opaque objects, but they are in general unable to capture dynamic scenes due to the sequential nature of

the acquisition process. Furthermore, materials that exhibit subsurface scattering, such as marble or skin, reduce the accuracy of the scanners, as light is not only reflected off the surface, but also penetrates it and reappears as a diffuse area of light rather than a point [Levoy et al., 2000]. Texture has to be captured by a different modality.

Time-of-Flight Time-of-flight (ToF) measurement acquires a dense depth map at each frame and is thus better suited for reconstructing dynamic scenes [Lange et al., 1999]. Time-of-flight sensors operate by emitting a light front towards the surface and recording its reflection at a sensor. The depth is then estimated from the flight time and the known speed of light [Iddan, 2001]. A variation of time-of-flight sensors use phase-modulation instead of a light front [Büttgen et al., 2005]. Time-of-flight sensors have various problems with inter-reflections or specular materials, where the emitted light might never return to the sensor or only returns indirectly after several bounces. Just as with laser scanners, texture needs to be captured by a different modality.

Structured Light Structured light methods project carefully designed patterns of light onto the surface, which are imaged by a camera. The captured light patterns encode the origins of the light rays, which enables depth estimation using triangulation. This information is either encoded temporally or spatially, or as a combination of both [Salvi et al., 2004]. Techniques that rely on temporal encoding, such as Gray-Code [Posdamer and Altschuler, 1982] or Phase-Shift [Scharstein and Szeliski, 2003], are less applicable to dynamic scenes. Techniques that employ spatial encoding, such as line [Zhang et al., 2002] or grid [Proesmans et al., 1996] methods, have low resolution and are less robust on textured surfaces. More recent techniques combine structured light with stereo to achieve more robust reconstruction [Zhang et al., 2003, Weise et al., 2007]. All structured light techniques have problems in areas where the surface occludes the projected pattern, which is often the case for a human face that undergoes wrinkling.

Passive Stereo Contrary to all of the previous methods, passive stereo does not have an emitting (active) component but operates solely with two or more sensing cameras. Depth is computed by triangulating corresponding points in the captured images. Passive stereo acquires both texture and depth in a single shot, is scalable and readily available — rendering the technique very appealing to dynamic surface reconstruction. Therefore this technique has drawn a lot attention and a broad spectrum of different

methods have been developed. Traditionally research has focused on the special case of two cameras (binocular stereo) [Scharstein and Szeliski, 2002] but with increasing availability of cameras and computing power, research has turned to general multi-view stereo (MVS) methods. Seitz et al. [Seitz et al., 2006] provide an excellent overview of the field and categorize MVS algorithms into four classes: 3D volumetric approaches [Hornung and Kobbelt, 2006, Son and Davis, 2006, Vogiatzis et al., 2007, Sormann et al., 2007, Kolev et al., 2009], surface evolution [Pons et al., 2005, Gargallo et al., 2007, Zaharescu et al., 2007, Delaunoy et al., 2008], computation and merging of depth-maps [Szeliski, 1999, Goesele and Curless, 2006, Strecha et al., 2006, Zach et al., 2007, Merrell et al., 2007, Campbell et al., 2008, Liu et al., 2009], and methods that grow surfaces from a sparse set of seed features [Goesele et al., 2007, Furukawa and Ponce, 2007, Jancosek et al., 2009, Hiep et al., 2009]. Most stereo methods aim to solve the problem in a general setting. These methods are not tailored to skin surface reconstruction and their accuracy is often far below of what can be achieved with active techniques, such as structured light. The technique we present in Chapter 5 on the other hand has been developed specifically to reconstruct human skin and achieves reconstruction qualities that are on par with active techniques. Our system combines stereo reconstruction with shape-from-shading to reconstruct a human face down to skin pore detail.

2.1.2 Normal Estimation

Contrary to methods that directly estimate the position of the surface, normal estimation techniques estimate its orientation. Once the orientation is known, the shape can be recovered through integration of the normal field. The two concepts discussed in this section (*photometric stereo* and *shape-from-shading*) have been implemented both with active and passive techniques.

Shape-from-Shading Shape-from-shading (SfS) describes techniques that estimate the orientation of the surface from shading properties. Since the first SfS technique developed by Horn in the 1970s [Horn, 1970], many different approaches have emerged [Zhang et al., 1999]. SfS computes the surface orientation from a single image based on intensity variations. This problem is ill-posed in general and thus the algorithms have to impose constraints on surface continuity, texture, color, reflectance and/or illumination.

Photometric Stereo Photometric stereo techniques recover surface orientation from multiple images of the same scene under varying illumination. The method was conceived in the 1980s by Woodham [Woodham, 1980]. Since then many different techniques have been proposed. These techniques typically require to capture the object under multiple controlled illumination conditions to produce accurate normal estimation for materials with unknown reflectance properties [Debevec et al., 2000, Gardner et al., 2003, Goldman et al., 2005, Chen et al., 2006]. Some methods reduce the amount of illumination conditions by imposing constraints on the reflectance properties [Lensch et al., 2003, Zickler et al., 2005, Hernández et al., 2008]. The cost of this tradeoff is that they miss some of the spatially varying effects of the surface. Ma et al. [Ma et al., 2007] avoids this by capturing the scene under varying polarization. Still they require four illumination conditions acquired under two different polarization states. Yet another approach was presented in [Glencross et al., 2008], who estimate a bas-relief from a single flash/no-flash image pair. The low hardware requirements and simple acquisition makes this an interesting option for texture acquisition.

2.1.3 Hybrid Techniques

The systems that provide the highest quality results to date make use of the complementary nature of both position and normal estimation.

Active The technique presented in [Nehab et al., 2005] to efficiently combine position and normal estimations has been utilized for faces in [Weyrich et al., 2006]. Ma et al. [Ma et al., 2007] presented a system for acquiring high-quality surface normals using polarized gradient-based illumination, which they combine with structured light scanning to generate high-resolution 3D reconstructions of human faces. The system is limited to a single viewpoint due to the polarization field, which is required to separate the diffuse from the specular reflections in order to estimate the surface normals. This limitation has been relaxed to some extent in [Ghosh et al., 2011], which employ several different polarization fields to allow the placement of the cameras freely on and up to 20° away from the equator. [Fyffe et al., 2011] also employ gradient-based illumination but circumvent polarization by estimating the diffuse-specular separation heuristically. All these methods require involved illumination setups and temporal multiplexing, which makes them less suited for performance capture. They either require high-speed cameras or register the frames to each other, which adds another layer of complexity to performance capture.

Passive The complementary nature of multi-view stereo (MVS) and shape-from-shading (SfS) has long been known [Blake et al., 1985]. Leclerc et al. [Leclerc and Bobick, 1991] use MVS purely as initialization for SfS. Torralba et al. [Torralba and Freeman, 2003] learn the relation between surface shading and coarse geometry from MVS and apply it to SfS to synthesize detail. As mentioned in Section 2.1.2, SfS is ill-posed and additional constraints on appearance and/or illumination are required. The albedo is usually constrained to be constant [Wu et al., 2011b], piecewise constant [Samaras et al., 2000, Jin et al., 2007], or smoothly varying [Fua, 1995]. The illumination is often assumed to be a single distant point light [Samaras et al., 2000, Jin et al., 2007]. Wu et al. [Wu et al., 2011b] recently introduced a technique that can cope with arbitrary illumination.

Considering that the strength of normal estimation is to recover small-scale surface variations and the strength of position estimation is to reconstruct the coarse shape, we propose a system to refine a coarse MVS reconstruction by minimizing an error function that integrates stereo, shading and smoothness terms. While this is similar in spirit to the method proposed by Fua et al. [Fua, 1995], we explicitly limit SfS to the spatial frequencies that are not captured by MVS. We call this approach *Mesoscopic Augmentation*, which allows us to reconstruct surfaces where albedo variations occur at a lower frequency than mesoscopic geometry variation. This is a reasonable assumption for human faces and allows for high-resolution reconstructions from a single exposure captured under constant illumination.

2.2 Hair Capture and Reconstruction

Driven by the difficulty of modeling hairstyles realistically by hand, various research has worked on capturing models of scalp hair. Grabli et al. [Grabli et al., 2002] introduced the idea of identifying 3D hair orientation from moving-light image sequences by the specular reflection peak and image-space orientation. Several other projects have since addressed the problem of modeling an entire head of hair from many views, with or without structured or controlled lighting [Paris et al., 2004, Wei et al., 2005a, Paris et al., 2008]. All these methods have focused on creating a model of the large-scale geometry, normally by constructing a smooth vector field in 3D that represents fiber orientation and then growing random fibers through this field. Because they focus on capturing whole-head hairstyles, capturing each hair would be prohibitive at the required scale. Recently Jakob et al. [Jakob et al., 2009] took a different approach, using depth-from-focus and

triangulation to capture the geometry of individual fibers in a smaller volume of hair. Their method requires hundreds of images and is thus not suited to capture live subjects.

Just like Jakob et al., the method presented in Chapter 6 reconstructs hair fiber by fiber, as it focuses on coupling the reconstruction of hair and skin in areas where both are present. This differs from traditional hair capture work with their goal of capturing whole heads of long hair. While Jakob et al. require hundreds of images to reconstruct the fibers, the proposed approach works with a single exposure, making it well suited to capture live subject.

2.3 Performance Capture

Data-driven facial animation has come a long way since the marker-based techniques introduced over two decades ago [Williams, 1990, Guenter et al., 1998]. In this section we review the related work on facial motion capture, starting with pure geometry approaches and techniques that fit parametric face models to images, followed by active lighting and marker-based techniques, and finally discussing more recent passive reconstruction methods.

Pure Geometry Approaches While image data, due to its superior resolution and detail, can be an enormous help in tracking 3D points over time, it is not always available, and sometimes just a sequence of 3D point clouds or incompatibly-triangulated 3D meshes are available as input. A number of authors have addressed the problem of tracking a 3D mesh over time based on pure geometry. An early approach was described by Anuar and Guskov [Anuar and Guskov, 2004]. They start with an initial template mesh that is then propagated through the frames of the animation, based on a 3D adaptation of the Bayesian multi-scale differential optical flow algorithm. Since this flow is invariant to motion in the tangent plane, it is not able to eliminate "swimming" artifacts during the sequence.

Inspired by the deformation transfer method of Sumner and Popović [Sumner and Popović, 2004], which allows to "copy and paste" mesh geometry from one shape to another, Winkler et al. [Winkler et al., 2008] present an optimization method to track triangle geometry over time combining terms measuring data fidelity, and preservation of triangle shape. Shape preservation is achieved using mean-value barycentric coordinates. This

also addresses motion in the tangent plane, eliminating the artifacts present in the results of Anuar and Guskov.

Tracking triangle geometry is intimately related to the problem of *cross-parameterization*, or *compatible remeshing* of shapes [Kraevoy and Sheffer, 2004], where the objective is to impose a triangle mesh representing one shape onto another, in a manner which minimizes distortion between the two. Bradley et al. [Bradley et al., 2008] have used this approach to track deforming garment geometry, however their method does not extend to faces since human skin does in fact distort during a performance.

More recent work by Sharf et al. [Sharf et al., 2008] tracks pure geometry over time using a volumetric representation. The main idea behind their method is that the volume of an object should be approximately constant over time, thus the flow must be "incompressible". This assumption, along with other standard continuity assumptions, regularizes the solution sufficiently to provide a good track. However, it is not applicable to facial performance tracking.

Fitting Faces to Images Another approach to performance capture is to start with a deformable face model (or "template") and then determine the parameters that best fit the model to images or videos of a performing actor [Li et al., 1993, Essa et al., 1996, DeCarlo and Metaxas, 1996, Blanz and Vetter, 1999, Pighin et al., 1999, Blanz et al., 2003, Vlasic et al., 2005]. Using this approach the approximate 3D shape and pose of the deforming face can be determined. However the deformable face tends to be very generic, so the resulting animations often do not resemble the captured actor. The face model must also be low-resolution for the fitting methods to be tractable, so it is usually not possible to obtain the fine details that make a performance expressive and realistic.

Markers and Active Light Probably the most common approach for performance capture is to track a sparse number of hand-placed markers or face paint using one or more video cameras [Williams, 1990, Guenter et al., 1998, Lin and Ouhyoung, 2005, Bickel et al., 2007, Furukawa and Ponce, 2009b]. While these techniques can yield robust tracking of very expressive performances and are usually suitable for a variety of lighting environments, the manual placement of markers can be tedious and invasive. Furthermore, the markers must be digitally removed from the videos if face color or texture is to be acquired. Also, marker resolution is naturally limited, and detailed pore-scale performance capture has not been demonstrated with this approach.

An alternative to placing markers on the face is to project active illumination onto the subject using one or more projectors [Wang et al., 2004, Zhang et al., 2004]. While this approach requires less manual setup, it can be equally invasive for the actor. Acquiring face color also poses a problem with these methods, as uniform illumination must be temporally interleaved with the structured light, sacrificing temporal resolution. A related active-light technique is proposed by Hernandez and Vogiatzis [Hernández and Vogiatzis, 2010], who use tri-colored illumination with both photometric and multi-view stereo to obtain facial geometry in real-time, however without temporal correspondence. Finally, combining markers and structured light with a light stage has proven to yield impressive facial performance capture results [Ma et al., 2008, Alexander et al., 2009].

Other researchers have recently used controllable lighting in a light stage setup for facial performance capture. Wilson et al. [Wilson et al., 2010] establish temporal correspondence for images under the changing light conditions of spherical gradient illumination. This allows them to combine stereo with photometric normal maps in a temporally consistent way to generate detailed facial geometry and performance capture. Current industry leading systems for active facial performance capture include Mova's CONTOUR Reality Capture[1], which requires fluorescent makeup. The makeup covers the face with a random pattern, allowing for higher-resolution tracking than with traditional dot-markers.

Passive Capture Recently, research has focused on passive reconstruction, without requiring markers, structured light or expensive hardware. Bradley et al. [Bradley et al., 2010] perform passive performance capture with automatic temporal alignment, but they lack pore-scale geometry and fail when confronted with expressive motions. Other passive deformable surface reconstruction techniques have been applied to faces [Wand et al., 2009, Popa et al., 2010]. Wand et al. [Wand et al., 2009] reconstruct from point cloud data by fitting a template model. However, their method tends to lead to loss of geometric detail which is necessary for realistic facial animation. This is partially resolved in recent work by Popa et al. [Popa et al., 2010] who use a gradual change prior in a hierarchical reconstruction framework to propagate a mesh structure across frames. Current industry leading systems for passive facial performance capture include Dimensional Imaging 3D[2].

[1] http://www.mova.com (accessed 27 June 2012).
[2] http://www.di3d.com (accessed 27 June 2012).

In the work presented in Chapter 7 we explore a novel concept called anchor-based reconstruction with robust image-space tracking, where similar facial expressions in a sequence form anchor frames that allow us to limit temporal drift, recover from tracking failure, match and reconstruct sequences in parallel, and establish correspondences between multiple sequences of the same actor captured on different occasions.

2.4 Optical Flow

Optical flow is the apparent 2D motion a point moving in 3-space traces on a 2D image plane. It provides a dense motion estimation for every pixel and is thus well suited to track deforming surfaces, such as human faces. Since the early work of Horn and Schunck [Horn, 1981] a great variety of different methods have been proposed. The middlebury database initiated by Baker et al. [Baker et al., 2011] gives a good overview of the available techniques. Most of these approaches rely in some way on the assumption that the intensity of a pixel (or the intensity gradient within a neighborhood) remains constant. For deformable objects, such as human faces, this assumption does not hold everywhere, i.e. in areas that undergo wrinkling. In the following section we will first review related work on motion tracking under brightness changes, and then discuss methods related to our illumination-aware geometry enhancement technique presented in Chapter 8.

2.4.1 Illumination-Robust Optical Flow

In the optical flow community several approaches have been proposed to cope with brightness changes in the input images. For purely additive brightness changes, imposing constancy of the image gradient is one solution, used for example in [Brox et al., 2004]. More sophisticated methods consider patch-based matching scores like cross-correlation [Molnar et al., 2010] that exhibit a greater degree of invariance, e.g. under multiplicative illumination changes. However, higher invariance discards some image information that could help the matching in image regions without brightness changes and sophisticated matching scores also complicate the implementation of optical flow algorithms. Even more involved are approaches that estimate spatially varying maps that explain multiplicative [Seitz and Baker, 2009] or additive and multiplicative brightness changes,

e.g. [Gennert and Negahdaripour, 1987]. This introduces additional unknowns that need to be estimated and regularized, which complicates implementation and introduces more tuning parameters that need to be carefully adjusted. Similarly, Haussecker and Fleet [Haussecker and Fleet, 2001] jointly estimate optical flow and model parameters of physical processes causing brightness change. However, only simple processes describing a moving illumination source and surface rotation are considered. If color images are available, using alternative color spaces like the HSV space can also improve robustness under illumination changes; see [Zimmer et al., 2011] and the references therein. However, using such color spaces might also cause problems, e.g. the H- and S-channels of the HSV space do not distinguish grayscales [Zimmer et al., 2011]. In practice, we also found no color space transformation that can adequately separate illumination changes from albedo, e.g. when skin wrinkles occur.

Instead of changing the optical flow algorithm to handle face tracking during expressive performances, we propose an alternative approach in Chapter 8, which leverages information about the 3D scene geometry to cancel out brightness changes. In this, we certainly tackle less general scenarios, but being tailored towards a specific application in mind, i.e. the temporal tracking of deforming surfaces, we achieve high accuracy results even using readily available optical flow algorithms.

Similar to our approach, Wedel et al. [Wedel et al., 2008] also modify the input images to ease flow estimation. They decompose the images into a structure and a texture part showing large scale image features and high frequency details, respectively. Then, mainly the texture part is considered for flow computation as artifacts like shadows should be mainly visible in the structure part. However, this only holds for larger objects casting shadows and not for small details like wrinkles, which are in the focus of our work. Moreover, the texture part also contains most of the noise and thus decreases the robustness of the flow estimation.

Part I

Acquisition and Calibration

CHAPTER 3

Acquisition

At the very beginning of every reconstruction pipeline stands data acquisition. The algorithms presented in the following chapters require images captured from various viewpoints. Acquiring the best possible data is essential for high-quality reconstruction results, as these will only be as good as the input data is to begin with. As described in Chapter 1 we require a setup that can capture the subject with very high spatial and temporal sampling. Phrased differently, the setup requires:

- **Fast Acquisition** - The data required for a reconstruction should ideally be acquired within a single short exposure. This allows to capture the fast movements that occur in human performances.
- **Full Coverage** - The setup should allow to capture the complete volume of interest at equally high quality.

One modality that exhibits these properties is *passive multi-view stereo (MVS)*. In addition to the above mentioned properties passive MVS incorporates several other positive features, the most important one being its passive nature. Passive in this context means that the scene is not augmented in any way to facilitate reconstruction. In particular, the setup contains no emitting components that would interact with the scene and no markers or other physical augmentation is required. This has the advantage that a simple illumination

setup suffices (Section 3.2) or that it could be combined with an active system that measures other properties such as appearance.

In this chapter we propose various hardware setups to capture MVS imagery, to synchronize cameras and to illuminate the scene.

3.1 Camera Setup

The algorithms developed in this thesis can operate with different camera configurations. In the following we will first present atomic two- and three-camera configurations, the building blocks for the other setups. Second we describe the constructed static setups consisting of an eight-camera setup suited to capture the complete face, as well as a 14-camera setup designed to capture facial hair. Lastly we introduce our dynamic setup suited to capture facial performances.

3.1.1 Camera

This section introduces camera design choices and settings used in the multi-camera configurations. Table 3.1 summarizes the acquisition hardware used during this thesis along with the preferred settings.

- **Resolution** - The camera should provide enough resolution such that the mesostructure of the skin (pores, freckles, etc.) are imaged. Those are required for matching. When reconstructing non-skin facial features such as hair, the features must be resolved well on the sensor. We use resolutions from 1 MP up to 20 MP.
- **Noise** - Noise has an adverse effect on the reconstruction quality and should be avoided as much as possible. It is best to use a camera with a big sensor (big pixels) and to capture at low ISO. We capture at ISO 100.
- **Lens** - Its best to image the full face such that it is centered on the sensor and covers most of it. Lenses with variable focus should be avoided if possible as they are typically of lower quality compared to fix-focal lenses. Longer focal lengths are better suited, as they exhibit less distortion. On the other hand they require a larger distance to the subject. We found that lenses between 50 and 100 mm offer a good compromise and employ both 60 mm and 85 mm lenses to capture the face and 100 mm to capture details.

	Canon 500D	Canon 550D	Dalsa Falcon 4M60
Pixel Size (μm)	4.68	4.17	7.4
Bits per pixel	14	14	10 (8)
Resolution (px)	4770 $\times$ 3168 (1584 $\times$ 2376)[1]	5184 $\times$ 3456 (2592 $\times$ 1728)[1]	2352 $\times$ 1728 (1176 $\times$ 864)[1]
fps	3.4	3.7	60 (42)[2]
f-Stop	11	11	11
Exposure	-[3]	-[3]	8 ms
Shutter type	mechanical	mechanical	global

Table 3.1: Camera specs - *This table summarizes the most important specs of the cameras that were used. ([1]) the effective resolution used is lower due to the Bayer pattern. ([2]) the effective fps count is lower because the cameras are operated in non-concurrent mode. ([3]) the exposure time is given by the illumination as described in Section 3.1.5.*

- **Depth-of-Field** - It is important that the face is imaged sharply. If the setup is to accommodate different people it is best to use a large depth-of-field to ensure that this is the case. The depth-of-field is controlled by the aperture size (f-stop). Increasing the f-stop increases the depth-of-field. Too small apertures (large f-stops) have an adverse effect, as diffraction effects are becoming apparent and the lens looses sharpness. Another point to consider is that a lens has typically the best imaging properties at two stops above the minimal aperture. We found empirically that f-stops from 8 to 11 provide a good compromise, giving sharp images over a reasonably large depth-of-field.
- **Exposure time** - The faster the motion within the scene the lower the exposure time has to be to avoid motion blur. For a person sitting still an exposure time of 1/100s is sufficiently low. If, on the other hand, the intent is to capture performances or fast motions (children, trembling person, etc.) then exposure times of 1/1000s or lower have to be chosen.

3.1.2 Atomic Configurations

The atomic configurations described in this section are configurations consisting of the least amount of cameras required for the proposed algorithms to work. They form the building blocks of both the static and dynamic setups.

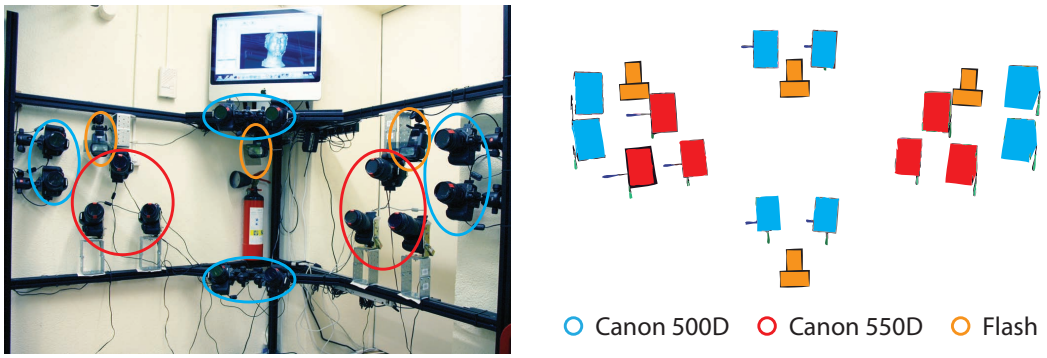


Figure 3.1: Static capture setup - *This Figure shows the fourteen-camera setup with flashes. Without the red cameras (Canon EOS 550D) the setup corresponds to the eight-camera setup. The setup utilizes the direct polarized illumination approach described in Section 3.2.2. Point lights (flashes) are used to illuminate the face and specular highlights are attenuated using cross-polarization.*

One-Camera Configuration While most of the algorithms presented in this thesis require a plurality of cameras to operate, some methods, such as *Mesoscopic Augmentation* in Chapter 5 or *Ambient Occlusion Cancelling* in Chapter 8, can operate on a single camera. A camera c captures an image $\mathcal{I}^c$. The origin of the image is at its upper left corner, with the x-axis pointing to the right and the y-axis pointing down. The coordinate frame of the camera was chosen such that the X- and Y-axes align with the x- and y-axes of the image and the Z-axis points towards the scene, spanning a right-handed coordinate system. Chapter 4 discusses the camera frame in more detail.

Two-Camera Configuration Two cameras are at least required to perform stereo, which is why we humans have two eyes. It is advisable to use two cameras of the same type equipped with the same type of lens. The two cameras should be oriented and positioned such that either their X- or Y-axes align and verged such that they capture the same area of interest. The spacing of the cameras is called the baseline and its size is a trade-off between good localization in the XY-plane and good localization along Z.

Three-Camera Configuration Two cameras are sufficient to reconstruct a two-dimensional surface, assuming the texture on the surface has variation in both dimensions. However, to reconstruct a one-dimensional structure such as a hair fiber in general configuration, three cameras are required. This

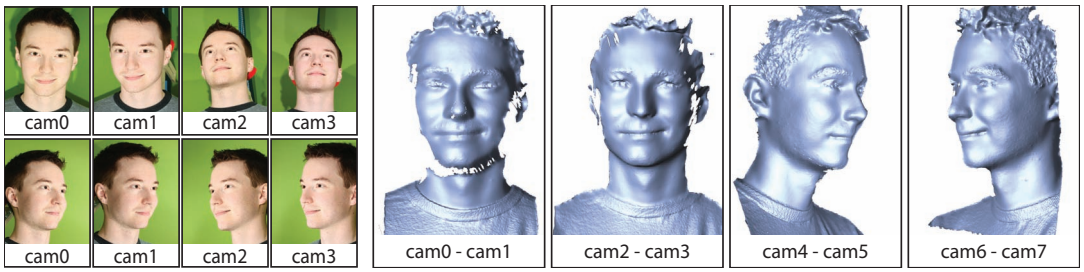


Figure 3.2: Sample dataset - *This figure shows a sample dataset captured with the eight-camera setup (Section 3.1.3) along with the corresponding 3D reconstructions of the individual tuples.*

phenomenon is called the aperture problem [Horn, 1981]. The three cameras are positioned roughly at the corners of an equilateral triangle to avoid pathological cases due to the aperture problem.

3.1.3 Static Setups

Static Eight-Camera Setup The eight-camera setup is composed of four camera pairs (tuples) that are distributed around the object of interest, a human face, to get the best coverage with respect to the features of the face. The setup comprises a tuple on each side to capture the left and right profile of the face, stacked vertically. These are the two most important tuples. A third tuple, arranged horizontally, captures the face from front and slightly from above, aiming at getting good resolution of the forehead and the back of the nose. The fourth tuple, also arranged horizontally, captures the face from below and mainly aims at covering the underpart of the jaw. Figure 3.1 shows the setup and Figure 3.2 displays a set of images captured with the setup along with the corresponding 3D reconstructions from the individual tuples.

The hardware used for this setup consists of 8 Canon EOS 500D equipped with 60 mm lenses. These cameras have a 15 MPixel CMOS sensor. This setup is employed to capture data for static face reconstructions performed by the algorithms presented in Chapter 5.

Static Fourteen-Camera Setup The fourteen-camera setup extends the eight-camera setup with cameras that focus on particular areas of the face. The setup was developed to capture facial hair (see Chapter 6) and the cameras therefore focus on the chin area. They are arranged as two camera

triplets and placed on each side of the face. The hardware used for this setup consists of 8 Canon EOS 500D equipped with 85 mm lenses and 6 Canon EOS 550D with 100 mm macro lenses. The Canon EOS 550D contains a 18 MPixel CMOS sensor.

3.1.4 Dynamic Setup

The dynamic setup is conceptually very similar to the static eight-camera setup. However, as we had only seven cameras at our disposal, the front tuple is replaced by a single camera. Even though the camera can not be used in pairwise stereo reconstruction it still contributes in the refinement stage (see Chapter 5). The hardware used in this setup consists of 7 Dalsa Falcon 4M60 equipped with 60 mm lenses. The Falcon 4M60 contains a 4 MP sensor and captures up to 60 fps at full resolution and concurrent readout. To avoid artifacts we operate the cameras in non-concurrent mode. Given the readout time of 16 ms and the exposure time of 8 ms, this leaves us with maximally 42 fps. The dynamic setup was used to capture data for Chapter 7.

3.1.5 Camera Synchronization

The Canon cameras are unfortunately not meant to be synchronized. Tests have shown that the highest degree of synchronization achievable with Canon EOS 500D cameras is in the order of 1/100s. The tests comprised triggering all cameras and a flash with a hardware trigger and observing whether all cameras captured the flash. This degree is not sufficient for fast moving objects (shutter speed 1/1000s) or when using flashes as illumination. We propose several approaches to overcome this problem, each with its own advantages and disadvantages.

Sequential Trigger & Flash The principle behind this approach is to open the shutters of all cameras and use the flash to synchronize them. As a consequence this only works in rather dark environments. The cameras are triggered in series using the Canon SDK, the camera which fires the flash is triggered last. The main advantage of this approach is that there is no special hardware required as the triggering happens via USB through the Canon SDK. Figure 3.3 demonstrates the principle. The minimum exposure time for the eight-camera setup is $\sim$ 1s and $\sim$ 2s for the fourteen-camera setup.

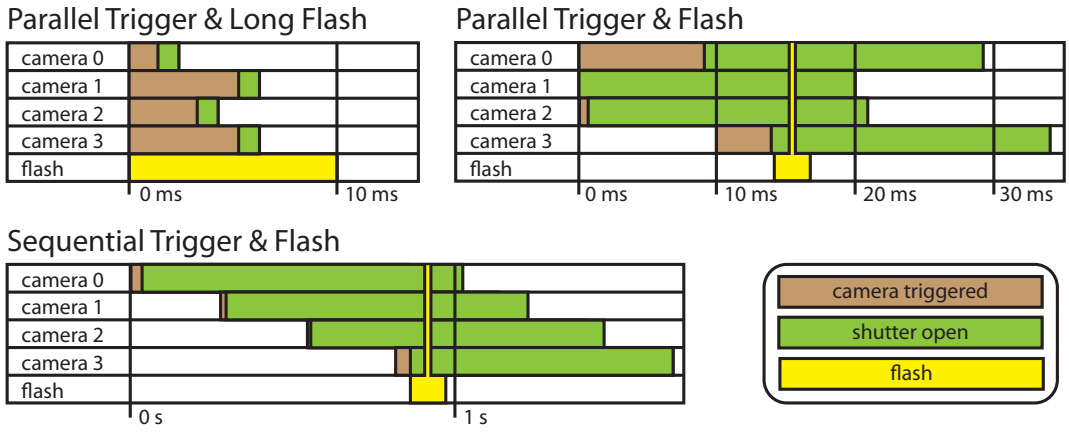


Figure 3.3: Camera synchronization - *Different approaches to synchronize consumer cameras that were not designed to be synchronized.*

Parallel Trigger & Flash This approach is related to the previous one. Again the shutters of the cameras are opened long enough so that the flash will be captured by all of them. This time however, the cameras are triggered in parallel through an external trigger. The important point is that the camera which fires the flash is triggered with a delay. Given the measured inaccuracy of 1/100s, we set the exposure times of the cameras to 1/50s and the delay to 1/100s. This approach works in scenarios with decent ambient light as well as with fast moving objects. The downside is that it requires custom made hardware.

Parallel Trigger & Extended Flash This approach is the inverse of the previous two. It requires an illumination device that is capable of holding its illumination over a longer period of time. LED panels are examples of such devices. The sequence in this approach is: turn on the light, trigger cameras, wait 1/100s plus camera exposure time, turn off the lights. This approach is not suited to capture fast moving objects.

3.2 Illumination

One major advantage of using a passive approach is that no specialized illumination is required. Specifically, we do not need to modulate the illumination over space and/or time. The face only needs to be well lit, ideally using uniform omnidirectional illumination. This section describes different illu-

mination setups that were constructed. They all have different advantages and shortcomings, which will be discussed as well. The main challenge is to avoid viewpoint dependent illumination effects such as specular highlights. These effects cause an object to appear differently in different viewpoints and aggravate or even impair the stereo matching process.

3.2.1 Indirect Illumination

In this approach all light that illuminates the face is bounced off one or more intermediate objects, called diffusors. This is the classical approach used in photography to avoid harsh shadows and specular highlights. Diffusors can either be transmissive (e.g. soft-boxes or light-tents) or reflective (e.g. bounce umbrellas) and typically a combination of both is used. All results in [Beeler et al., 2010] and Chapter 5 were generated using indirect illumination. The main disadvantage is the bulky setup which requires a lot of space making this method only suitable for studio setups.

3.2.2 Direct Polarized Illumination

This approach illuminates the face directly from several point-lights (flashes), which produces less even illumination of the face than the previous setup but has no additional space requirements. To avoid specular highlights, which would have a negative influence on the stereo reconstruction we make use of cross-polarization. Linear polarizers are mounted in front of the flash and (rotated by 90°) the lenses. This effectively blocks light that is directly reflected at the surface of the skin as this light maintains its polarization state. Light which enters the skin becomes (partially) depolarized and is thus not fully blocked by the polarization filter [Ma et al., 2007]. Figure 3.1 shows the setup. One disadvantage of polarization filters is that they block most of the light ($\sim$ 75%) and are thus not energy efficient. Ordinary flashes are capable of producing sufficient light for a split-second, making this approach well suited for compact static setups. All results in [Beeler et al., 2012] and Chapter 6 were generated using direct polarized illumination. For dynamic setups, where the face is captured at high frame-rates, ordinary flashes are not to be recommended as they need to recharge after each flash. Strobe lights would be a viable option in such a scenario or direct omnidirectional illumination, as described in the next section.

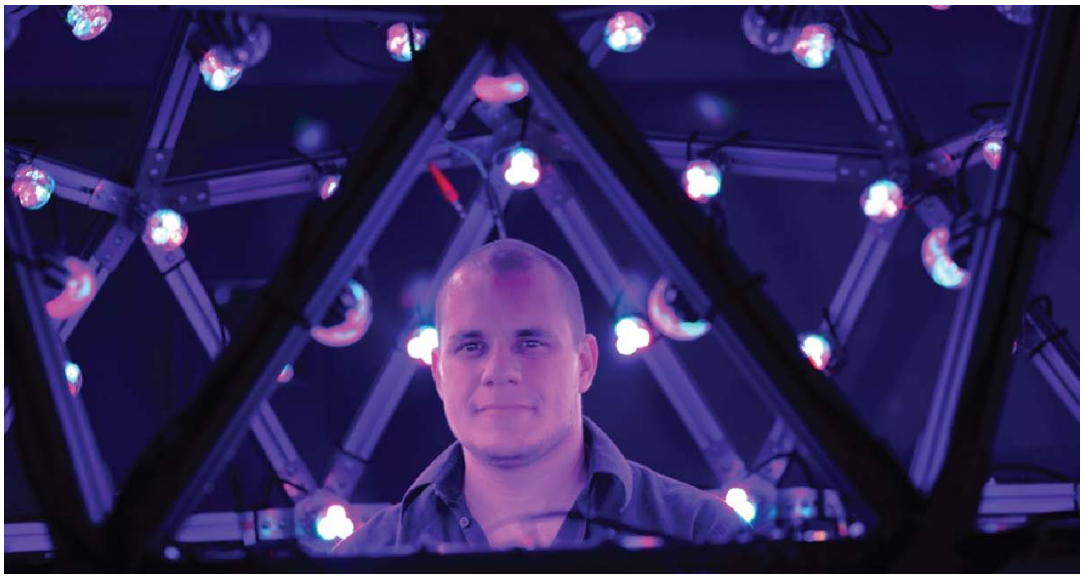


Figure 3.4: Helios *is a versatile multi-purpose light stage, which can produce arbitrary spatio-temporal illumination.*

3.2.3 Direct Omnidirectional Illumination

Specular reflection is not bad per-se, the problem are spatially varying specular reflections (highlights). The face can be lit directly if done so uniformly from all directions. A spherical light stage approximates this model well, as it arranges a great number of point light sources spherically around the face. All results in [Beeler et al., 2011] and Chapter 7 were generated using direct omnidirectional illumination. Even though a simple configuration using static white light-sources would suffice for the algorithms presented in this thesis, we decided to develop a versatile multi-purpose light stage, called Helios.

3.3 Helios — Multi-purpose Light Stage

Helios was designed as a versatile and multifunctional light stage (Figure 3.4). It consists of 468 LEDs arranged in 156 RGB LED triplets. Each LED is individually controllable with up to 12bit intensity resolution allowing to display spatially and spectrally varying illumination scenarios. Furthermore, the illumination scenarios can be changed with up to 1000 Hz, which allows to rapidly acquire a sample under many different illumination conditions.

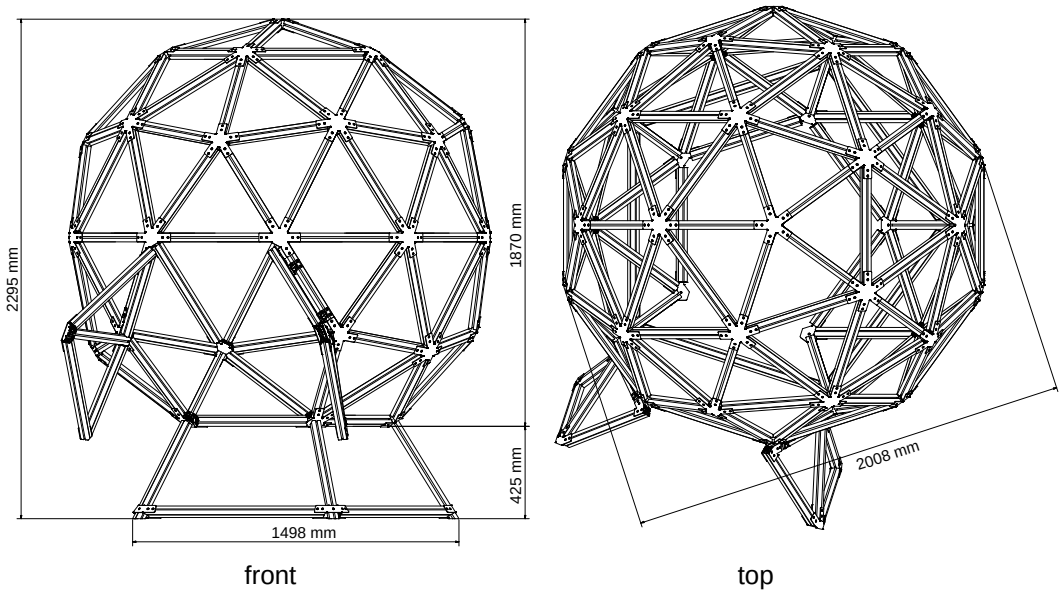


Figure 3.5: Structure - *The structure of helios is a once subdivided icosahedron with a diameter of 2 meters. The bottom of the icosahedron has been removed and replaced by a socket, such that the center of the sphere is with approximately 1.3 meters over ground at the position of the head of a sitting person. To facilitate access to the interior of the dome we added a door on one side.*

The light of all LEDs is focused onto the center of the light stage, illuminating the sample absolutely evenly at up to 30 kLux. In the following we will first describe the physical structure of the light stage, and then address its electronics.

3.3.1 Structure

As basis for the light stage we chose an icosahedron which we subdivided one time at the edges. This structure is also referred to as 2V/L2 icosahedron geodesic dome[1]. The complete 2V/L2 icosahedron has 42 vertices, but we removed the bottom vertex as shown in Figure 3.5. We also added a door to facilitate access to the interior of the dome. The final structure consists of 41 vertices (connectors), 115 edges (struts) and 75 triangular faces. Because of the subdivision, the polyhedron is no longer regular, as can be seen in

[1] http://simplydifferently.org (accessed 27. June 2012).

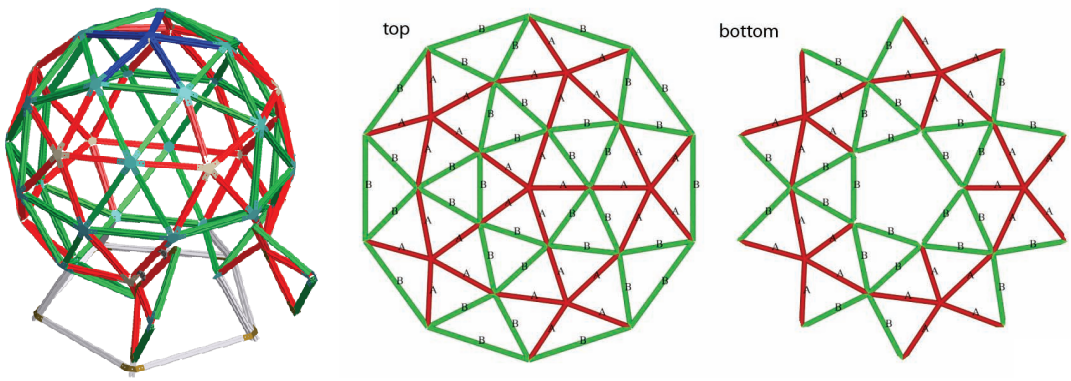


Figure 3.6: Topology - *Due to the subdivision, the resulting polyhedron looses the regularity of the icosahedron. This figure shows the distribution of A (red) and B (green) trusses.*

Figure 3.6. The dimensions of the dome have been chosen with a radius of 1 meter such that it can accommodate a sitting person.

Connectors The 41 vertices connect either 6 struts (25x) or 5 struts (16x). Figure 3.7 shows the design for a 6-way connector. A connector consists of two metal plates attached both from the inside and outside of the light stage to increase stability.

Struts The 115 struts form the edges of the icosahedron and exist in two types. Struts of type A (55x) have a length of $\ell_A \approx 0.55$ cm and form an angle of $\beta_A \approx 15.86°$ at the vertex. Those of type B (60x) are $\ell_B \approx 62$ cm long and form an angle of $\beta_B \approx 18°$ at the vertex. See Figure 3.8 (b) for a schematic. To facilitate mounting the lights and other devices we chose standard strut profiles with a thickness of $\rho = 4$ cm.

Faces The 75 triangular faces also exist in two different versions due to the subdivision. The 20 equiangular faces are formed by three B struts, enclosing angles of 60° at the vertices. The other 55 faces are formed by two A and a B strut, enclosing angles of $\alpha_A \approx 55.57°$ and $\alpha_B \approx 68.86°$, resp. See Figure 3.8 (a) for a schematic.

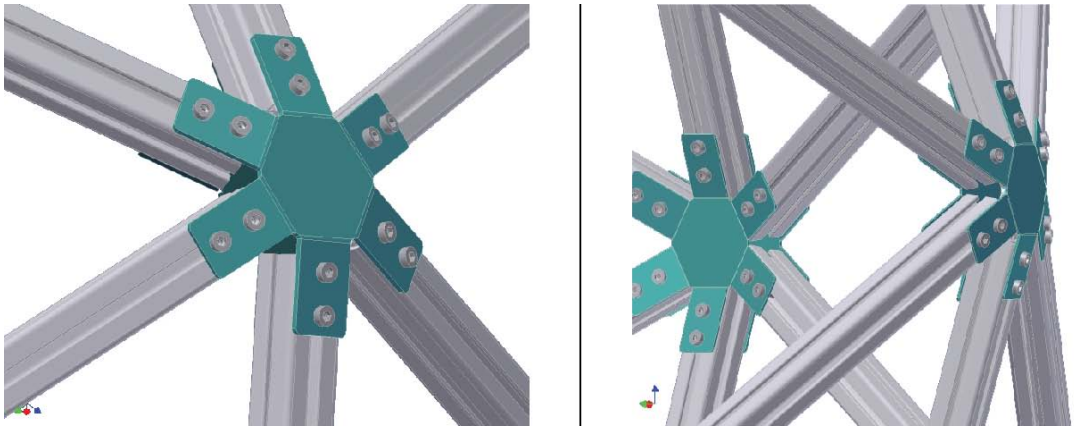


Figure 3.7: Connectors - *The connectors at the vertices of the dome were manufactured from aluminum plates, which are attached to the trusses both from the inside and the outside to increase stability.*

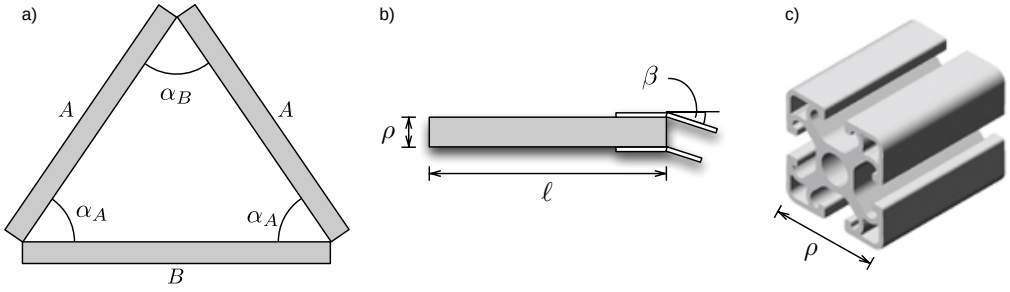


Figure 3.8: Struts - *(a) Irregular face formed by two A and a B strut. The A struts are with $\ell_A \approx 0.55$ cm shorter than the B struts ($\ell_B \approx 0.52$ cm) which results in different angles $\alpha_A \approx 55.57°$ and $\alpha_B \approx 68.86°$. (b) β denotes the angle that is formed at a vertex and differs again for A and B struts ($\beta_A \approx 15.86°$ and $\beta_B \approx 18°$) struts. (c) To facilitate mounting the lights and other devices we chose standard strut profiles with a thickness of $\rho = 4$ cm.*

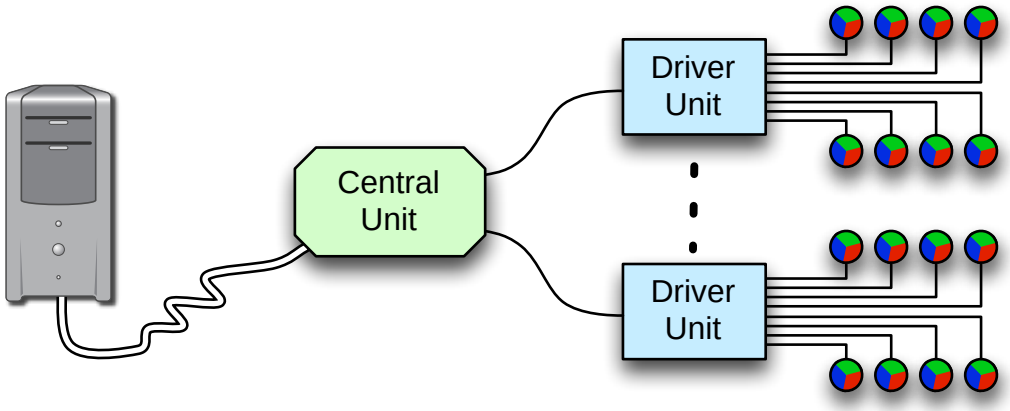


Figure 3.9: Connection scheme - *From right to left: Up to eight light triplets are controlled by a driver unit. The 20 driver units are connected via fiber optics to the central unit. The central unit handles communication with the controlling computer via USB and routes incoming commands to the driver units.*

3.3.2 Electronics

Figure 3.9 shows an overview of the connection topology. The 468 LEDs are arranged in 156 RGB LED triplets. Each triplet is equipped with red, green and blue LEDs (Philips Lumileds LUXEON K2) as well as a 45° lens and a heat sink. Figure 3.10 shows an LED triplet along with its specifications. The light stage produces a total of ∼34'000 Lumen, which converts to ∼28'000 Lux within the working volume at the center of the dome. To achieve this power ∼2.4 kilo Watts are required, which are provided by a TDK-Lambda Genesys Power Supply. Despite the strong current (∼157 Ampere) operation is nonhazardous because the units operate at low Voltage (15 V). The 156 triplets are mounted at the vertices (41x) and at the center of each strut (115x) and are oriented towards the center of the light stage. Light control is organized hierarchically via *driver* and *central units*. The hierarchical organization has several advantages, one of which is that the cable length can be kept short to reduce electromagnetic radiation.

Driver and Central Units Each of the 20 driver units (Figure 3.12) controls up to eight LED triplets. The intensity of the LEDs is controlled via *Pulse Width Modulation (PWM)* as shown in Figure 3.11. The relative amount of

Color	Wavelength	Half-width	Current	Voltage
Red	627 nm	20 nm	700 mA	3.60 V
Green	530 nm	35 nm	1000 mA	3.72 V
Blue	470 nm	25 nm	1000 mA	3.72 V

Figure 3.10: LED light unit - *A light unit is equipped with red, green and blue LEDs (Philips Lumileds LUXEON K2) as well as a 45° lens and a heat sink. The table lists specifications for the used LEDs. The unit consumes approximately 8 Watts at full operation and produces 200-240 Lumen.*

time during which the LED is turned on (T_P^{on}) vs. off (T_P^{off}) within the duration of a pulse (T_P) defines its intensity.

$$I = \frac{T_P^{on}}{T_P} \tag{3.1}$$

The amount of quantization steps within a pulse define the dynamic range of the LED. The shortest quantization step is inversely proportional to the frequency of the PWM (32MHz). Given a fixed quantization step, the dynamic range can be controlled by increasing the duration of a pulse. On the other hand, the amount of frames per second that can be displayed (temporal resolution) depends directly on the length of the pulses, as a frame requires at least one pulse. Bit-depth of the dynamic range and temporal resolution of the dome are therefore related according to

$$f_P = \frac{1}{\frac{1}{32e6}2^n}, \tag{3.2}$$

where $f_P = 1/T_P$ is the pulse frequency, n the desired bit depth and $32e6$ is the core frequency of the PWM. For 15 bits, the target pulse frequency lies at approximately 1kHz. For 16 bits at about 500Hz. Trading bit-depth versus pulse frequency has its limits. On the one side the pulse frequency should remain high to avoid flicker (at least twice the capture speed). On the other side the LED drivers require approximately 150μs until the are fully established [TexasInstruments, 2012], which introduces an upper limit to the pulse frequency as can be seen from Figure 3.13.

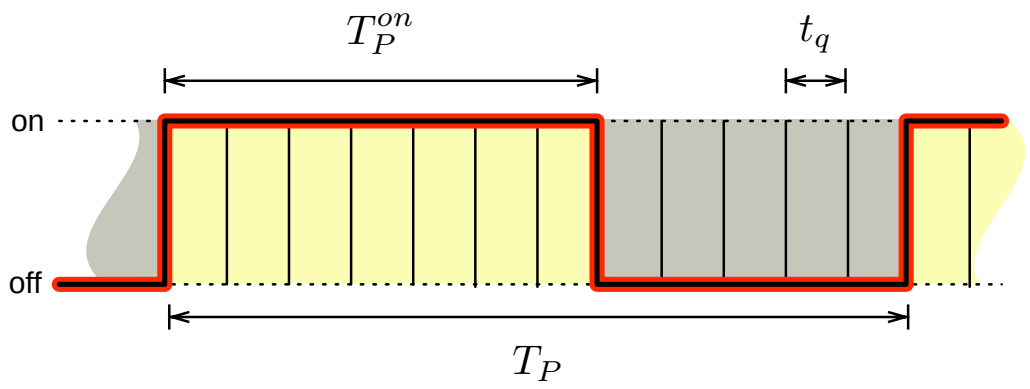


Figure 3.11: Pulse width modulation (PWM) - *The intensity of the LEDs is controlled via pulse width modulation. The relative amount of time an LED is turned on T_P^{on} during a the duration of a pulse T_P defines its intensity. The dynamic range (intensity resolution) of the LED depends on the amount of quantization steps t_q within a pulse.*

The unit is further equipped with persistent memory of 512 kB to store up to 10922 illumination scenarios, which allows the light dome to run autonomously without the controlling computer once the desired illumination scenarios have been uploaded. Every driver unit can in addition generate a trigger signal through which cameras and other equipment can be synchronized to the light stage. This is essential when capturing temporally varying illumination patterns. The driver units are mounted on the dome close to the light triplets it controls to minimize cable length. The driver units are connected in a star topology to the central unit via fiber optics. The central unit handles communication with the controlling computer via USB and routes incoming commands to the driver units. It is also issuing commands to change illumination scenarios, keeping the individual driver units synchronized.

3.4 Discussion

Acquiring the best possible data is essential for high-quality reconstruction results, as these will only be as good as the data is to begin with. The choice of camera, lens, camera arrangement, depth-of-field or illumination greatly influence the quality of the data, to name just a few factors. In this chapter we discussed different camera configurations and settings, we proposed

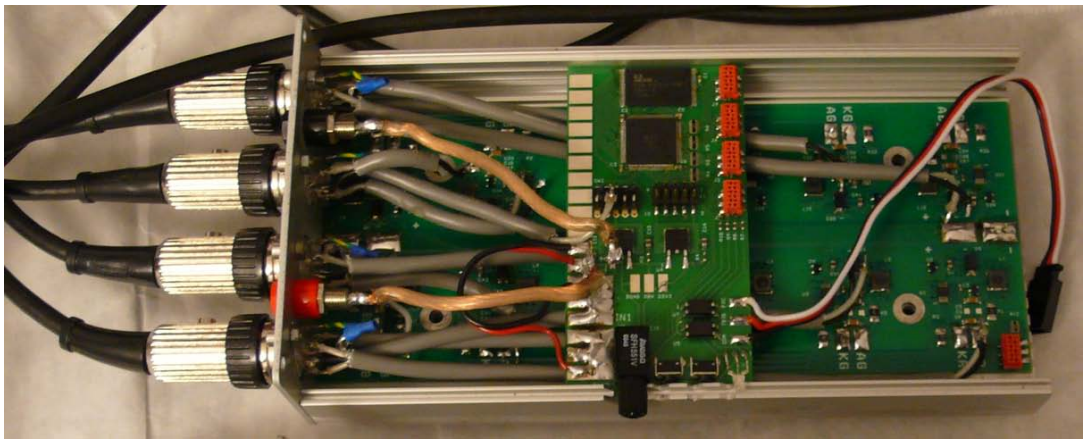


Figure 3.12: Driver unit - *The circuit boards of the driver units were custom designed in-house and produced by an external company. The final assembly and cabling of the devices was again done in-house.*

methods to synchronize the cameras and to illuminate the subject and we gave a complete overview over the different setups that were used during the course of the thesis. Furthermore, we described Helios, a versatile multi-purpose light stage. While designed and constructed as part of this thesis, the light stage was developed with future applications in mind. In this thesis the dome was employed purely as a source of static illumination in Chapter 7.

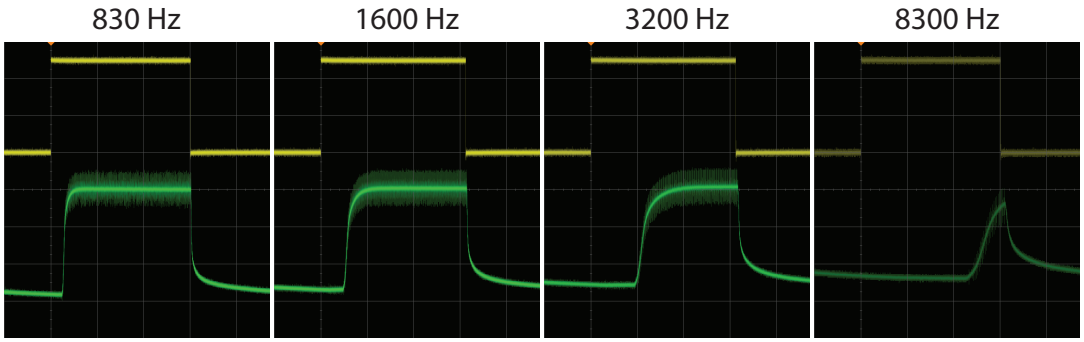


Figure 3.13: LED driver lag - *The LED driver (lower signal in green) requires approximately 150 µs to fully respond to the input signal (upper signal in yellow). This lag effectively reduces the dynamic range for higher frequencies.*

CHAPTER 4

Calibration

Calibration is the task to determine unknown properties of a system. For 3D reconstruction, accurate calibration is key to producing high-quality results. It is therefore essential to ensure that the system is always well calibrated. At the same time, calibration is often a time-consuming and tedious process, and therefore avoided whenever possible. In this chapter we present calibration methods that are tailored towards face capture and offer accurate and very user friendly calibration. We start with camera calibration and then discuss techniques for light stage calibration.

4.1 Camera Calibration

When calibrating cameras two aspects are important — geometric and radiometric calibration. Geometric calibration is the task to find the camera extrinsics (its position and orientation in space) as well as its intrinsics, such as focal length. These properties are described in more detail in Section 4.1.1. Radiometric calibration relates to how incident illumination is transformed to pixel values.

4.1.1 Camera Model

In this thesis we use a pinhole camera model [Hartley and Zisserman, 2000] and assume that the lens is distortion free. Given the hardware used (see Chapter 3), this model is a simple and sufficient approximation. A point $\mathbf{X}$ in world space ($\mathbb{R}^3$) is projected onto a pixel $\mathbf{p}$ in image space ($\mathbb{R}^2$) by multiplying $\mathbf{X}$ in homogeneous coordinates with a projection matrix P, which is given as

$$P = K\,[R|\mathbf{t}]\,. \tag{4.1}$$

The *extrinsic calibration* of the camera, given by rotation R and translation $\mathbf{t}$, defines how a point in world coordinates is transformed to camera coordinates. The *intrisic calibration* matrix K defines how a point in camera coordinates is projected onto the image plane and converted to pixel coordinates. Assuming that the camera sensor is rectangular and consists of square pixels, the calibration matrix K contains three degrees of freedom; the focal length f and the principal point $\mathbf{c} = (c_x, c_y)$ in pixel coordinates. The intrinsic calibration matrix is written as

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.2}$$

For many modern cameras, such as the ones used in this thesis, it is reasonable to assume that the principal point $\mathbf{c}$ is at the center of the camera sensor, reducing K to a single unknown f.

4.1.2 Geometric Calibration

The theoretical foundation of camera calibration is well established [Zhang, 2000, Tsai, 1987] and our focus has been on the practical matter of achieving a straightforward and reliable calibration for a face-capture system. The practical considerations and requirements for the calibration method are:

- **Accurate** - The calibration method should be as accurate as possible to allow for high-quality reconstructions.

- **Automatic** - User interaction required by many calibration applications [1], such as clicking corners, should be avoided. The method should work fully automatically.
- **Robust** - The calibration should still work if only part of the calibration target is captured, as it is often difficult or even impossible to position an object such that it is fully imaged by all cameras in a setup.
- **Fast Acquisition** - Unlike previous multi-camera calibration techniques that require several hundred images [Svoboda et al., 2005] the method should work with a minimal set of images. The proposed method operates on a single image per camera.
- **User Friendly** - The required calibration target should be inexpensive and easy to produce. Performing the calibration should be straightforward and convenient.

The practical assumptions and constraints that we use to alleviate the problem are derived from the intended use to calibrate face-capture setups:

- **Camera Configuration** - The cameras are arranged concentrically around a small volume of interest where the face will be situated. The calibration must be accurate for this volume only.
- **Sensor Quality** - We assume the cameras are equipped with CMOS or CCD sensors with square pixels, which is the case for most digital cameras. These sensors exhibit no skew and it is reasonable to assume the center of projection lies in the center of the sensor. This reduces the 5 unknowns of the intrinsic calibration matrix K introduced in Section 4.1.1 to 1, the focal length f.
- **Lens Quality** - We assume the lenses exhibit only little distortion. This holds for most DSLR fix-focal lenses which typically exhibit distortions in the order of $0.1\% - 0.4\%$[2].
- **Low Perspective** - The cameras should not exhibit large perspective distortions, meaning that the capture distance should be large compared to the focal length.

These assumptions hold for all the setups presented and used in this thesis. Cameras or lenses that do not conform with these assumptions can still be used by first calibrating their intrinsic parameters by a standard approach and then employ the proposed method to calibrate for the extrinsics of the complete setup.

[1] http://www.vision.caltech.edu/bouguetj/calib_doc/ (accessed 27. June 2012).
[2] http://www.dxomark.com/index.php/Lenses (accessed 27. June 2012).

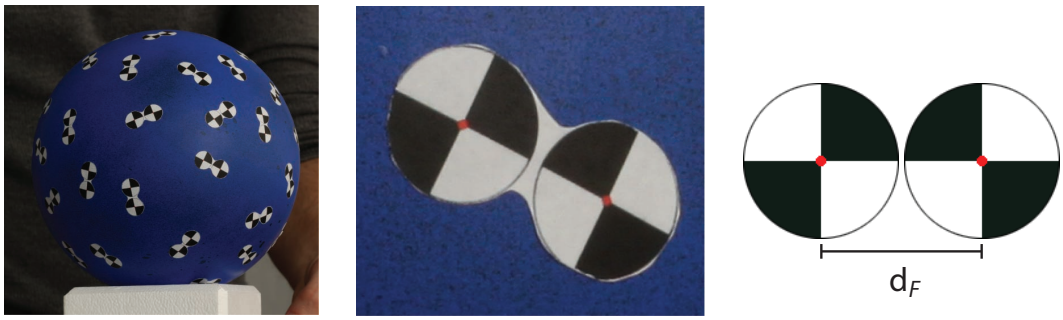


Figure 4.1: Calibration sphere - *Fiducials are placed randomly on the calibration sphere. A fiducial consists of two checkerboard circles with red dots at their centers. The fiducial defines a known distance d_F.*

4.1.2.1 Calibration Sphere

A roughly spherical object will serve as calibration target. The choice to use a sphere amounts from the fact that a sphere is equally fair to all view-points and the projection of a sphere onto the image plane will be roughly circular given the assumed imaging properties. The size of the calibration target is chosen to match the size of a human head and the calibration sphere occupies the same workspace as the subject's head in the run-time system. Thus, we ensure that calibration data is collected — and the calibration is therefore well-estimated — in exactly the same workspace as will be used at runtime.

The sphere is colored in a distinct color (we chose blue) to facilitate segmentation. The sphere is augmented with fiducials as shown in Figure 4.1. Each fiducial is a double circle. The center points of the circles provide the correspondences between cameras, as well as a known metric distance d_F that can be used to set scale. The fiducials are not used to provide known 3D coordinates — hence the sphere need not be perfect, and the fiducials can be placed by hand with arbitrary distribution. Fiducials were printed on sticky paper, and the slight distortion in fixing a flat sticker to a spherical surface was not found to be a problem.

4.1.2.2 Algorithm

The calibration process is fully automatic. The algorithm consists of the following steps:

SEGMENT CALIBRATION SPHERE
As described in Section 4.1.2.1, the calibration sphere is designed to appear predominantly in a distinct color. This is used to segment the sphere from the background. The image I^c of every camera $c \in \mathcal{C}$ is converted from RGB to HSV color space. Given the target hue of the sphere h_s, the filter response G is computed as

$$G(\mathbf{p}) = 0.5\sqrt{(\cos(H(\mathbf{p})) - \cos(h_s))^2 + (\sin(H(\mathbf{p})) - \sin(h_s))^2}, \quad (4.3)$$

where H is the hue channel of the HSV image. If the background itself already consists of a distinct color, which is the case in many studio setups (green screen), then the segmentation can also be used to segment the background instead of the sphere. Problems may arise if the background color is desaturated (e.g. shades of white and black). For these cases, the hue channel is ill defined and through inter-reflection might even faintly adapt the color of the sphere itself. In such cases it is advantageous to also use the saturation channel S as

$$G'(\mathbf{p}) = G(\mathbf{p})\, S(\mathbf{p}). \quad (4.4)$$

Both filter responses are defined within $[0, 1]$. Next we fit a circle to find the projection of the sphere. One common way to detect circles in an image is to use the circular Hough transform [Forsyth and Ponce, 2002]. We tried the openCV [3] implementation and found that it did not work robustly as the sphere is not perfectly spherical and the projection is thus also not perfectly circular. A much more robust way is to use an active contour [Forsyth and Ponce, 2002] to find the circle, as it can leverage additional information. As active contour we chose a circle $C(\mathbf{p}_c, r_c)$ with center $\mathbf{p}_c$ and radius r_c that maximizes the signed difference between its interior and exterior.

$$\underset{\mathbf{p}_c, r_c}{\text{maximize}} \quad \frac{1}{r_c - \delta_r} \oint_{C(\mathbf{p}_c, r_c - \delta_r)} I(\mathbf{p}) d\mathbf{p} - \frac{1}{r_c + \delta_r} \oint_{C(\mathbf{p}_c, r_c + \delta_r)} I(\mathbf{p}) d\mathbf{p}, \quad (4.5)$$

[3] http://sourceforge.net/projects/opencvlibrary/ (accessed 27. June 2012).

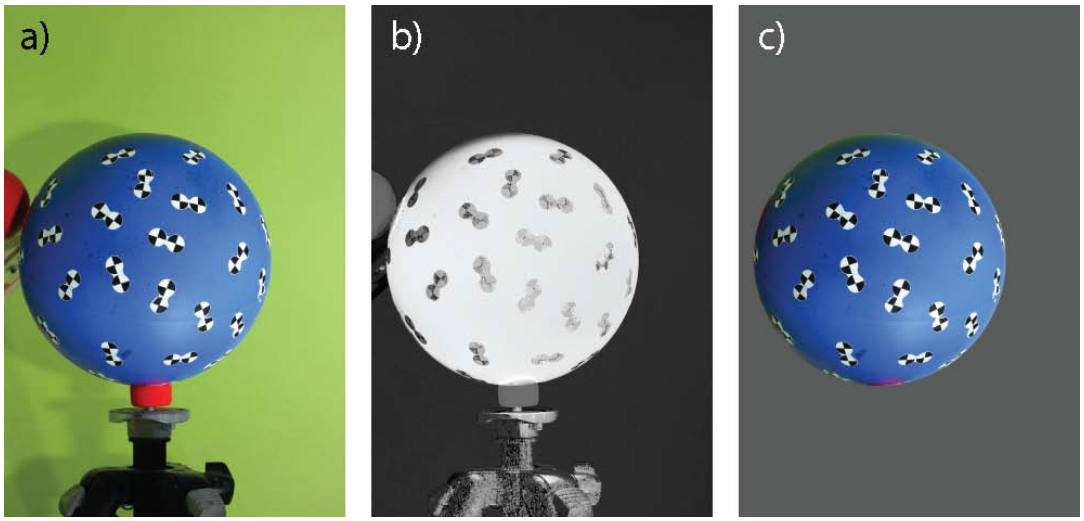


Figure 4.2: Detect projection - *(a) Input image of the calibration sphere. (b) Color filtering enhances the contrast between the calibration sphere and the background. (c) Segmented calibration sphere using active contour.*

where δ_r is set to 1 pixel. To be efficient and robust to varying projection sizes we employ a hierarchical scheme. Figure 4.2 shows an input image (a) color filtered (b) and segmented (c).

RECONSTRUCT CALIBRATION SPHERE

Once we have identified the projection of the sphere in a camera, we can reconstruct the sphere relative to this camera. To do so we require the known radius ρ_s of the sphere as well as a rough estimate of the focal length, given either by an educated guess or by consolidating the EXIF meta-data of the imagery.

The center $\mathbf{x}_s$ of the sphere lies on a ray $\mathbf{r}_s$ from the camera center $\mathbf{x}_c$ through the center of the projection circle

$$\mathbf{x}_s = \mathbf{r}_s(t_s) = \mathbf{x}_c + t_s \mathbf{d}_s, \tag{4.6}$$

where $\mathbf{d}_s$ is the normalized direction of the ray $\mathbf{r}_s$.

A ray $\mathbf{r}_i$ through the contour of the circle will be tangent to the sphere, as depicted in Figure 4.3. The parameter t_s can then be computed using basic trigonometry as

$$t_s = \frac{\rho_s}{\sqrt{1 - \langle \mathbf{d}_s, \mathbf{d}_i \rangle^2}}, \tag{4.7}$$

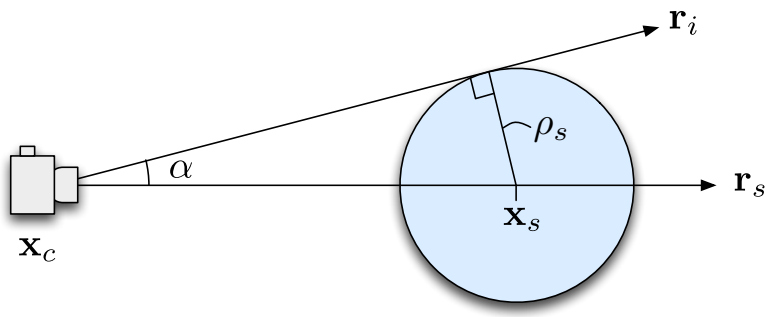


Figure 4.3: Reconstructing the calibration sphere - *The center $\mathbf{x}_s$ of the sphere lies on a ray $\mathbf{r}_s$ from the camera center $\mathbf{x}_c$ through the center of the projection circle. A ray $\mathbf{r}_i$ through the contour of the circle will be tangent to the sphere.*

where $\mathbf{d}_s$ and $\mathbf{d}_i$ are the normalized directions of the rays $\mathbf{r}_s$ and $\mathbf{r}_i$ and their inner product relates to the angle α in Figure 4.3. As a last step we apply a change of coordinate system and shift the origin of the coordinate frame to the center of the sphere. The choice to put the origin of the coordinate frame at the center of the sphere will play an important role in the fiducial matching stage of the algorithm described further down.

RECONSTRUCT FIDUCIALS

The fiducials are detected using the small colored dot in the center. The image I is converted to HSV color space and the response G is computed with Equation 4.3. The response is convolved with a LoG (Laplacian of Gaussian) filter kernel. The width of the kernel is computed from the projected diameter of the colored dot. Figure 4.4 (b) shows the response of the convolution. The n positions with the highest response are selected as initial fiducial candidates. The image is unwrapped in the neighborhood of each fiducial candidate using the reconstructed calibration sphere and the position of the fiducial is refined using the function *findCornerSubPix* from [Intel, 2012], which uses the checkerboard pattern to estimate the fiducial point with sub-pixel accuracy. The refined fiducial candidates are re-projected onto the reconstructed sphere to get estimations of their 3D coordinates. As can be seen from Figure 4.1 fiducials always come in pairs of two with known distance d_F between their centers. In addition, the checkerboard markings of the fiducials are mirrored, through which the orientation of a double fiducial is determined. These two properties permit to filter the set of fiducial candidates and robustly identify a set of oriented double fiducials.

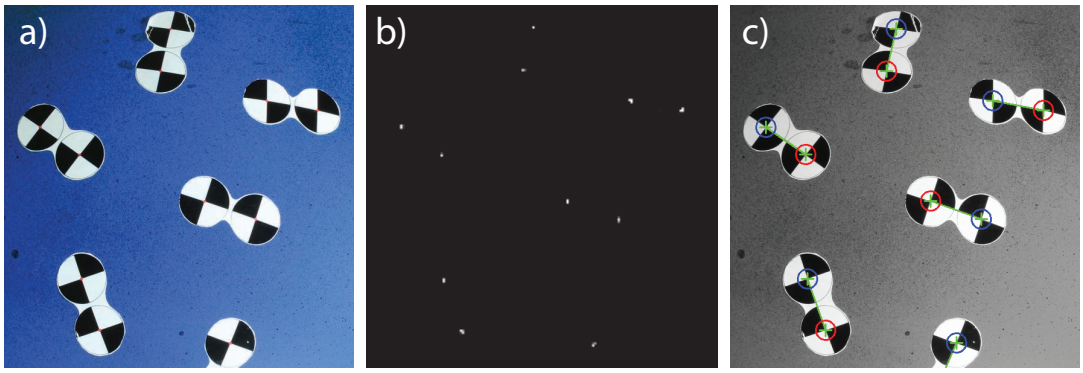


Figure 4.4: Fiducial detection - *(a) Close-up of the calibration sphere showing a few fiducials. (b) Response to the filters applied in Section 4.1.2.2. (c) Detected double fiducials consisting of an ordered pair of features shown in blue and red, resp.*

Match Fiducials

Each double fiducial reconstructed in the previous step forms an oriented triangular facet with the center of the sphere, as depicted in Figure 4.5 (c). Because the origin of the coordinate frames for both cameras is chosen to be at the center of the spheres, a pure rotation is sufficient to align the two coordinate frames. The rotation can be computed from a single match. As the fiducials themselves do not encode any identity, the matching relies on the global distribution of the fiducial tags. Any two double fiducials may be used to compute a hypothesis for the rotation, which is then verified using the remaining double fiducials. The hypothesis with the lowest matching error is kept. The matching error is given by the sum of squared differences in Euclidean space for all fiducials. This is essentially a RANSAC [Forsyth and Ponce, 2002] algorithm with a sample size of one.

The best match for every camera pair is computed and dynamic programming is employed to transform all cameras relative to each other. This gives an initial estimation of the overall calibration, which is refined in the next step.

Structure From Motion

Structure from Motion is a well established process to jointly reconstruct the scene geometry (Structure) as well as the camera calibration (Motion). The joint optimization is known as Bundle Adjustment (BA) [Lourakis and Argyros, 2009]. In a general setting, Bundle Adjustment requires a large number of features to work reliably — many more features than provided by the marker tags on the calibration sphere. However, the

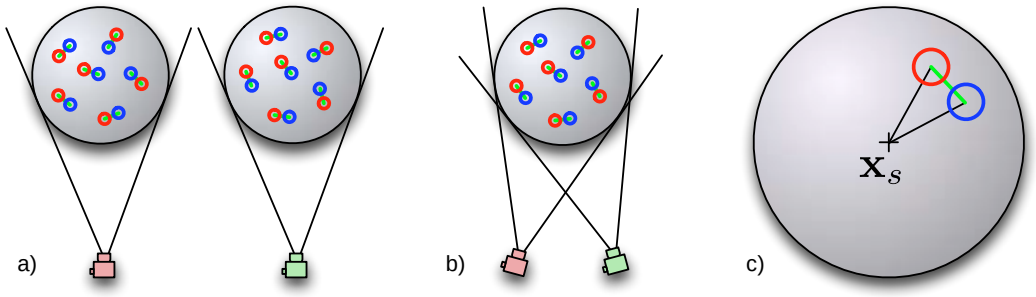


Figure 4.5: Feature matching - *(a) Given calibration spheres and fiducials for two cameras, the goal is to find a transformation of the coordinate frame that aligns the two spheres (b). Due to the choice of coordinate frame the transformation is a simple rotation, which can be computed from two correspondences. (c) These two correspondences are given by a single double fiducial, which consists of two ordered features (red, blue). Identity is not encoded within a double fiducial itself but through the global distribution of features on the sphere.*

previous stage already provides a good initial estimate of the overall calibration, enabling Bundle Adjustment to refine the solution and preventing it from getting stuck in a local minimum. Additionally we can benefit from the reduced degrees of freedom provided by the assumptions presented in Section 4.1.2. All intrinsics, except the focal length f, are kept fix during Bundle Adjustment. Therefore the result of this step is a Euclidean coordinate frame, which is accurate up to scale.

RECOVER SCENE SCALE

The last step is to establish scale to provide a metric coordinate frame. Scale is computed from the known distance d_F of the marker tags, adjusted to consider the curvature of the sphere

$$d'_F = 2r_S \sin\left(\frac{d_F}{2r_S}\right). \tag{4.8}$$

The scale factor is then

$$s = d'_F \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \hat{d}_{F_i}, \tag{4.9}$$

where $\hat{d}_{F_i}$ is the current unscaled distance of fiducial F_i and $\mathcal{F}$ is the set of

all fiducials. Once the scene is properly scaled we compute a 3D 'capture-zone' as the volume occupied by the reconstructed 3D fiducial points plus some margin, to delimit the region within which 3D processing will be done at run-time.

The so computed calibration is most accurate on the surface of the calibration sphere, where the fiducials are located. The quality degrades slowly when moving away from this surface. This can lead to reconstruction artifacts when face and calibration target do not line up. To improve reconstruction quality, we can refine camera calibration during reconstruction using Bundle Adjustment on features computed from the face itself — therefore altering the calibration to be most accurate on the surface of the face. This is discussed in more depth in Chapter 5.

4.1.3 Radiometric Calibration

A camera converts pixel irradiance to intensity values, which are stored as an image. The conversion function is called *Camera Response Function (CRF)* or *Camera Transfer Function (CTF)*. Most image processing algorithms operate directly on intensity values. The algorithm presented in Chapter 8 however operates directly on pixel irradiance, which requires the CRF to be undone. In general, CRFs are non-linear and must be recovered empirically (calibrated) in order to find their inverse function. CRF calibration is usually performed by capturing a grayscale chart [Brady and Legge, 2009] or imaging a scene under several exposures [Debevec and Malik, 2008]. Digital cameras employ the CRF as a post-process, as the sensors themselves (CMOS or CCD) exhibit a linear response [Debevec and Malik, 2008]. Raw, unprocessed camera images are therefore typically linear and can be used without additional transformation.

4.2 Light Stage Calibration

Just like the camera calibration discussed in Section 4.1 the light stage has geometric and radiometric properties that are unknown. See Chapter 3 for a description of the constructed light stage. For the methods described in this thesis, the light stage needs not to be calibrated as it is only required as a source of static omnidirectional illumination. For completeness and because of its value for other use-cases of the light stage, we describe in this section an algorithm to calibrate the geometric properties of the light stage.

The geometric properties of the light stage — its position $\mathbf{x}_\ell$ and orientation $\mathbf{o}_\ell$ — are calibrated using a mirror sphere. We found that bearing balls are great mirror spheres, as they are made of metal and are almost perfectly spherical with high reflection coefficient (since their surface is smoothly polished to minimize friction).

As a preprocessing step we compute the 3D positions of the individual lights of the light stage in a local coordinate frame using the known geometry of the light stage. This greatly simplifies geometric calibration, reducing the task to finding the relative position $\mathbf{x}_\ell$ and orientation $\mathbf{o}_\ell$ (6 DoF) of the complete light stage instead of having to determine these properties for each individual light (6n DoF).

The geometric calibration now consists of the following steps:

- Detect light reflections
- Detect mirror sphere
- Reconstruct mirror sphere
- Estimate light stage position and orientation

4.2.1 Detect Light Reflections

The proposed algorithm to detect reflections of individual lights on the mirror sphere is designed based on the following observations and assumptions:

1. The difference in intensity between direct and indirect reflections is high
2. Light reflections of different lights do not overlap
3. The difference in area covered by the light and its reflection on the mirror sphere is large

Observation 1 allows to apply a threshold to classify pixels that are illuminated. These pixels are either part of a direct reflection on the mirror sphere or of a light itself, as can be seen in Figure 4.6 (a). These two cases can be distinguished based on the area they cover (observation 3). Following observation 2 the reflections do not overlap and we can therefore extract the area they cover using region growing with hysteresis [Canny, 1983]. Applying a threshold on the size of the area separates the pixels belonging to a direct reflection from others caused by capturing the light itself. The output of the algorithm are clusters of pixels belonging to individual light reflections on the mirror sphere. Note that assumption 2 would be violated if the lights are

distributed very densely on the light stage. In such a case, assumption 2 can be satisfied by operating only on a subset of the lights at a time and repeating the algorithm for every subset.

4.2.2 Detect Mirror Sphere

As the mirror sphere reflects the environment, it is itself almost invisible as can be seen from Figure 4.6 (b). Detecting it using the circular Hough transform [Forsyth and Ponce, 2002] or the active contour approach presented in Section 4.1.2.2 is therefore not an option. We thus designed an algorithm that detects and reconstructs the mirror sphere indirectly through the shape of the light reflections on the mirror sphere. The reflections are detected as described in Section 4.2.1. Capturing all three color channels at the same time would violate assumption 2 as the spectra of the different LEDs overlap. To avoid crosstalk of the individual color channels we capture and process the three colors separately. The algorithm to detect the mirror sphere is based on the observation that the shape of a small area light varies as a function of the distance of the reflection from the projection of the center of the sphere, as can be seen in Figure 4.6 (a).

Using principal component analysis (PCA) on the area covered by the reflection we compute its ellipticity. As can be seen from 4.6 (a), the ellipticity of the reflection is a function of the distance of the reflection from the projected center of the sphere. Furthermore, the major eigenvector is tangential to the concentric iso-contours of the projection of the sphere, which implies that the minor eigenvector intersects the projected center of the sphere. Computing intersections of the minor eigenvectors from the reflections with high ellipticity gives a good initial guess of the projected center of the mirror sphere (see Figure 4.6 (b)).

4.2.3 Reconstruct Mirror Sphere

Before we can estimate the position and orientation of the light stage we need to reconstruct the position of the mirror sphere relative to the previously calibrated cameras. This is achieved by triangulating the projections of the centers detected in the previous step. Given the know diameter of the sphere the mirror sphere can be reconstructed in space relative to the cameras. Figure 4.6 (c) shows the re-projection of the reconstructed sphere onto the original image in blue.

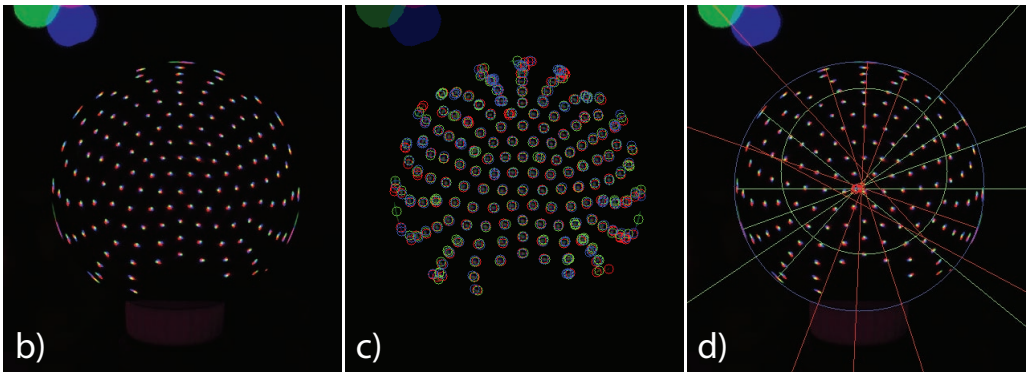


Figure 4.6: Geometric light stage calibration - *(a) Image captured of the mirror sphere reflecting all lights of the light stage (small colored dots) as well as a few lights in the background. Note how the ellipticity of the reflections increases when moving away from the projected center of the sphere. (b) Detected reflections overlaid over the original image. (c) Reflections with large ellipticity are used to estimate the center of the sphere. The blue circle depicts the projection of the reconstructed mirror sphere.*

4.2.4 Estimate Light Stage Position and Orientation

Position $\mathbf{x}_\ell$ and orientation $\mathbf{o}_\ell$ of the light stage are calibrated using three specific lights. The lights are chosen to be approximately at mutually orthogonal positions on the light stage as seen from the center. Using spectral multiplexing we can image all three lights at the same time by choosing the first light to be red, the second green and the last one blue. For every camera $c \in \mathcal{C}$ the reflections of the three lights are detected using the algorithm described in Section 4.2.1. Given these reflections and the mirror sphere reconstructed in Section 4.2.3, reflection rays $\mathbf{r}_i^c$ are computed for all three lights $i \in \{R, G, B\}$. A reflection ray $\mathbf{r}_i^c$ is the re-projection ray for the detected reflection in camera c, reflected at the mirror sphere s . Following the fact that all reflection rays of a given light should originate from the same point in space $\mathbf{x}_i$ we solve the non-linear optimization problem

$$\begin{aligned} \underset{\mathbf{x}_\ell, \mathbf{o}_\ell, \mathcal{T}}{\text{minimize}} \quad & \sum_{i \in \{R,G,B\}} \sum_{c \in \mathcal{C}} \left| M_{\mathbf{x}_\ell, \mathbf{o}_\ell} \mathbf{x}_i - \mathbf{r}_i^c(t_i^c) \right|^2 \\ \text{subject to} \quad & t_i^c \geq 0, \; i \in \{R, G, B\}, \; c \in \mathcal{C} \end{aligned} \tag{4.10}$$

to find the position $\mathbf{x}_\ell$ and orientation $\mathbf{o}_\ell$ of the light stage. $M_{\mathbf{x}_\ell, \mathbf{o}_\ell}$ is the Euclidean transformation matrix given by the position $\mathbf{x}_\ell$ and orientation $\mathbf{o}_\ell$ of the light stage and t_i^c are the line parameters. There is a line param-

eter for every light for every camera, so the optimization problem solves for $6 + 3\,|\mathcal{C}|$ unknowns. While a single camera would provide enough constraints to estimate position and orientation of the light stage, the algorithm makes use of all available cameras and solves the over-constrained system with Levenberg-Marquart.

4.3 Discussion

Accurate calibration of the acquisition device is essential to produce good results. Inaccurate calibration can lead to noisy results or prevent reconstruction completely. The contributions presented in this chapter are twofold. The first contribution is a practical camera calibration technique designed specifically for face capture. It allows to quickly, effortlessly and accurately calibrate a face-capture setup, as it does so completely automatically from a single exposure per camera. This greatly reduces the time effort spent on calibration and encourages to calibrate the setup often to ensure proper calibration. While the calibration is accurate within the calibrated head-size volume, it does not necessarily extrapolate well — making the method less suited to calibrate large volumes. One possible extension would thus be to capture the sphere at several positions in the larger volume and then merge the individual calibrations.

The second contribution is a technique to geometrically calibrate the lights in a light stage. The method is designed to calibrate our multi-purpose light stage Helios presented in Chapter 3. The technique requires only a few images of a mirror sphere captured with pre-calibrated cameras. Radiometric calibration of the light dome is outside the scope of this thesis but is an essential topic for future work.

Part II

Geometry

CHAPTER 5

Skin Surface Reconstruction

As discussed in Chapter 1, capturing a high-quality model of a human face is of great interest in multiple domains — for the movie and games industries, in medicine, to provide more natural user-interfaces, and for archival purposes. All these domains are in need of accurate high resolution 3D geometry of human faces.

The current method of choice for this task is an active system based on laser, structured light or gradient-based illumination. Active light brings robustness because it effectively augments an object surface with known information. On the other hand, it requires special-purpose hardware and often employs time-multiplexing. Polarization-based methods further constrain deployment to a single camera at a fixed viewpoint. Contrast this with passive stereo vision, which has the potential to be an extremely versatile modality for constructing 3D models — it captures in a single shot, readily adapts to different arrangements and numbers of cameras with no constraint on camera position, seamlessly integrates 3D data captured over multiple distances and at different scales in a scene, captures texture that is intrinsically registered with the recovered 3D data, and uses commodity hardware. However, in the past, the reliability and accuracy of passive stereo have fallen short of what is available from active systems, and it has not been used for capturing high-quality face models.

In this chapter we propose a passive stereo vision system that computes the 3D geometry of the face with reliability and accuracy on a par with a laser scanner or a structured light system. We introduce an image-based embossing technique to capture mesoscopic facial geometry[1], so that the quality of synthesized faces from our system equals that achieved with gradient-based illumination. In practical terms, we equal the performance of active systems while attaining the advantages of passive stereo already listed, particularly capture in a single shot under standard light sources, low-cost, and ease-of-deployment. To demonstrate the robustness of the system, we show results for faces of varying gender, ethnicity and age. To demonstrate versatility, we show face models captured off a studio setup and off a consumer binocular-stereo camera, with the latter result suggesting that 3D face scanning is poised to move beyond the professional arena and become a practical application on the desktop.

Even though the system is tailored to capture continuos surfaces such as human skin, it is robust in areas that do not comply with these assumptions, such as eyes, and provides reasonable results. To extend the field of application, Chapter 6 improves this system to permit coupled reconstruction of skin surface and sparse facial hair.

5.1 Reconstruction Pipeline

This section describes the end-to-end system as shown in Figure 5.1. Camera calibration is a pre-processing stage and is described in Chapter 4. The run-time system begins with pairwise stereo matching, and uses a pyramidal approach in which results at lower-resolutions guide the matching at higher-resolutions as described in Section 5.1.2. At each layer of the pyramid, matches are computed at pixel level to give dense matches across the face, and the matches are used to generate a 3D mesh as described in Section 5.1.4. The mesh is refined using a modification of the traditional approach, in which photo-consistency and smoothing terms are augmented with a novel term that captures fine detail at the pore-level. This is described in Section 5.2. An excellent overview and categorization of MVS is found in [Seitz et al., 2006]. Optionally, as described in Section 5.1.3 the system can improve the camera calibration during the reconstruction process to correct for inaccuracies in the calibration and improve the reconstruction results.

[1] We use the term mesoscopic for geometry at the scale of pores and fine wrinkles, and macroscopic for overall 3D shape of the face.

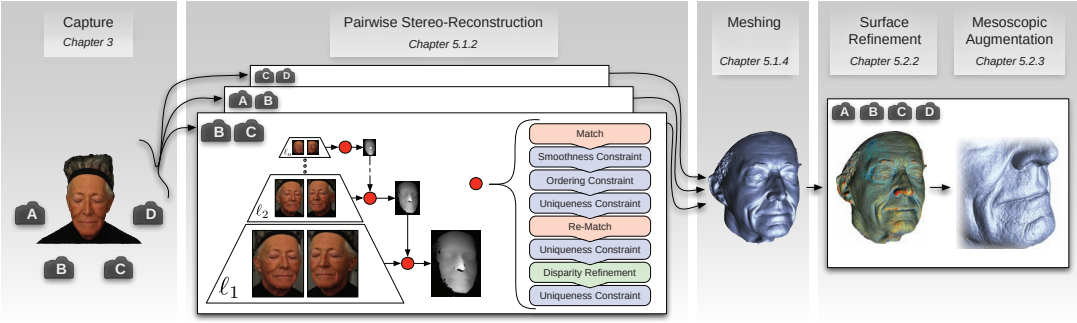


Figure 5.1: The proposed system - *The subject is captured with multiple cameras. This figure shows a four-camera setup, but the system can incorporate an arbitrary number of cameras.*

5.1.1 Image Preprocessing

The images are retrieved in 12 bit RAW format from the cameras and converted to floating point RGB images. After de-Bayering using VNG [Chang and Cheung, 1999], the images are subsampled once by a factor of two due to the Bayer-pattern. Subsampling reduces the data size and attenuates sensor noise. A mask is automatically generated using cues of color hue and saturation. The reasoning is that in the chosen setup the face of our subject forms the biggest connected patch of color with high saturation. For this, the image is first transformed to HSV color space and the binary mask M is then computed as

$$M(x,y) = \begin{cases} 1 & S(x,y)\left(\pi - |H(x,y) - h|_\alpha\right)/\pi > \beta \\ 0 & \text{otherwise} \end{cases}, \tag{5.1}$$

where h denotes the hue of human skin, the norm $|\cdot|_\alpha$ the absolute angular difference $[0,\pi]$, H the hue $[0,2\pi[$ and S the saturation $[0,1]$. The domain of the threshold β is $[0,1]$ and we use $\beta = 0.2$ in all our examples. Only the largest connected region in M is retained and holes in this region are filled up.

The images are finally reduced to grayscale by retaining only the green channel.

5.1.2 Pairwise Stereo-Reconstruction

In this section we describe the individual steps of our stereo reconstruction. Matching is done pairwise between neighboring cameras[2], and at pixel level to establish dense matches across the face. For a given camera-pair, the first step is to rectify the images to obtain row-aligned epipolar geometry. An image pyramid is generated for each rectified image by factor-of-two subsampling using Gaussian convolution. The image resolution at the lowest-resolution layer of the pyramid is chosen to be around 150×150 pixels, but this is approximate and the criteria is simply that the major facial features are still visible.

Each layer of the pyramid is then processed as follows: First, matches are computed for all pixels as described in Section 5.1.2.1. Next, we check smoothness, uniqueness and ordering constraints for each pixel (see Section 5.1.2.2). Pixels that do not fulfill these constraints are re-matched using a limited search area (Section 5.1.2.1). The limited search area ensures that smoothness and ordering constraints hold on the re-matched pixels. The uniqueness constraint however needs to be enforced once more. The disparity maps are then refined. An in-depth description is deferred to Section 5.2.1, since it is an instantiation of the refinement formulation introduced in Section 5.2. Finally, the uniqueness constraint is enforced again.

Matching starts at the lowest-resolution layer of the pyramid. The resulting disparity map provides input to the the next higher layer, where it is used to constrain the search area for matching, and so on up to the highest-resolution layer of the pyramid. As demonstrated in Figure 5.2, this leads to a hierarchical refinement of the reconstruction over the layers of the pyramid.

5.1.2.1 Pixel Matching

Following the taxonomy of [Scharstein and Szeliski, 2002], the system employs a winner-take-all block-matching algorithm using normalized cross-correlation (NCC) as matching cost over a square window (3x3). Matching is performed along the epipolar line only, which is why we represent a pixel coordinate by a scalar value that denotes its position on the epipolar line. Pixel p in image I is matched against all pixels in image J within a given search area and the best match is retained. The disparity at p is computed to subpixel accuracy by computing NCC values for p against the matching pixel q

[2] Pairing of cameras in a multi-camera system is done manually, although it would be straightforward to automate if needed.

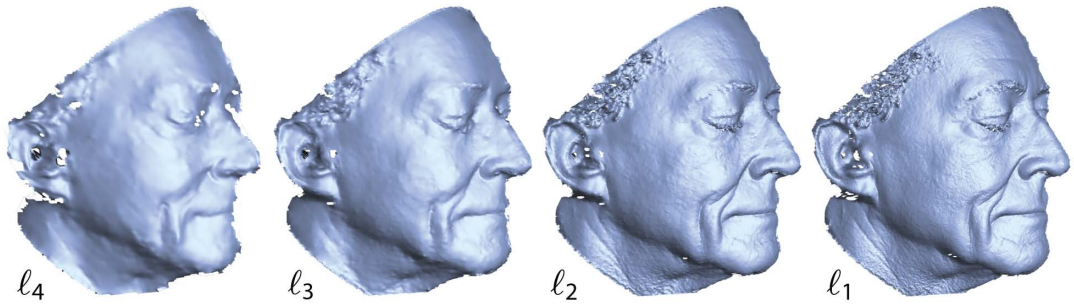


Figure 5.2: Hierarchical reconstruction - *Reconstructions of a stereo-pair at 4 different layers of the pyramid starting from the coarsest layer ℓ_4 (160px $\times$ 160px). The dimensions double at each layer up to the highest resolution layer ℓ_1 (1280px $\times$ 1280px).*

and its two neighbors in image J, fitting a polynomial of degree three, and finding the position in image J where it is at minimum.

Matching is performed twice per layer. The initial matching computes putative matches for all pixels using the disparity estimates of the preceding layer (or the 'capture-zone' if no preceding guesses are present) to constrain the search area. Next, we check for each pixel smoothness, uniqueness and ordering constraints (see Section 5.1.2.2). Pixels that do not fulfill these constraints are re-matched using the disparity estimates of the neighboring pixels that fulfilled the constraints to limit the search area.

5.1.2.2 Constraints

The system can make use of constraints that hold for human faces in the given setting. Pixels in image I are matched against image J, and vice-versa from image J to image I. Acceptance of a match at pixel p in image I is subject to three constraints -

- **Smoothness Constraint** - computed disparity at p is consistent with neighbors in a surrounding window. In our implementation this is achieved by enforcing that more than half of all neighbors in a 3×3 neighborhood differ by a disparity less than one pixel.
- **Uniqueness Constraint** - the matching needs to be bijective: if p in image I matches to q in image J then q must also match to p. To take different foreshortening into account we tolerate a disparity mismatch of up to one pixel in our implementation.

- **Ordering Constraint** - computed disparity at p does not exceed the disparity of its right-neighbor pixel by more than one pixel.

5.1.3 Improving Camera Calibration

Sometimes the calibration provided is not perfect. This might be because calibration was not performed accurately or some of the calibration parameters deviated over time. The algorithm described in this section may help in such a case to improve reconstruction results. Following the idea presented in [Furukawa and Ponce, 2009a] we apply a bundle adjustment [Lourakis and Argyros, 2009] step after the MVS reconstruction presented in the previous section to improve the camera calibration. We do so by generating a random set of samples ($\sim$ 10000) from the generated disparity maps. The samples are triangulated to 3D points and projected into all cameras. Pixel matching (Section 5.1.2.1) on a larger square window (7x7) is applied to find better correspondences within a search range for all cameras. Bundle adjustment is used to compute a better guess of the camera calibration from these correspondences. To remedy the influence of mismatched samples, we triangulate the 3D samples a new and compute their reprojection error in all views. Samples with high reprojection error are discarded and bundle adjustment is repeated on the remaining sample set. If the improved calibration differs from the original one, the previous MVS reconstruction is discarded and computed again using the new calibration. This algorithm may be applied several times, either until convergence or for a predefined number of iterations. A small number of iterations ($\sim$ 4) is usually sufficient. While this approach is very helpful when suboptimal camera calibrations are available, it was not applied to generate any of the results in this chapter.

5.1.4 Meshing

Each camera-pair in Section 5.1.2 produces one disparity map, which is used to compute a corresponding array of 3D points and a corresponding array of surface normals. Since we estimate a dense disparity map, the normals are computed using finite differences on the points. 3D points and surface normals are collected across all camera pairs. Outliers are removed using a simplified approach of [Merrell et al., 2007]. If two 3D points project onto the same pixel in a given camera view, both with normals facing towards that camera, and without an intermediate point with normal facing away from the camera, then the associated topology is incorrect, and the 3D point with the higher matching error is rejected (see Figure 5.3 for visualization). The

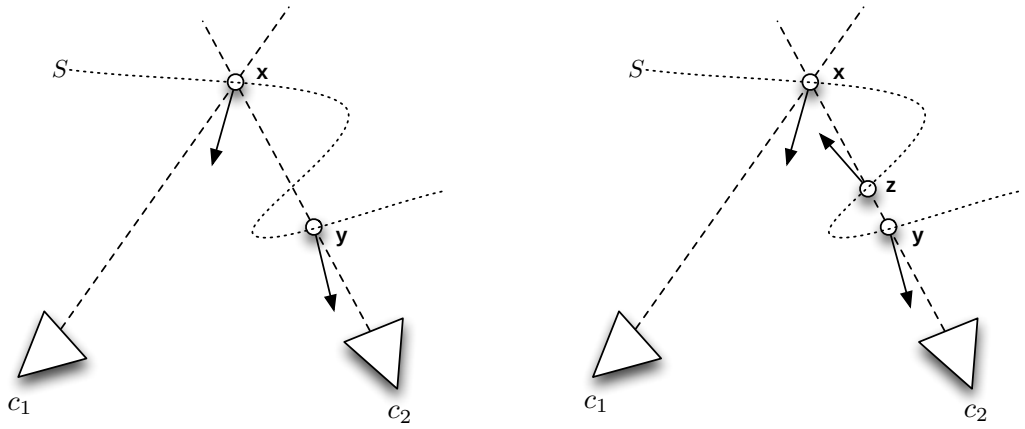


Figure 5.3: Outlier filtering - *Left: Point* **x** *is in conflict with Point* **y**, *since both project to the same point in* c_2 *and since no surface S can match this constellation. The point with higher matching error will be rejected. Right: Point* **x** *does not conflict with Point* **y**, *since the constellation can be explained by a valid surface S due to Point* **z**.

resulting set of 3D points and normals is input to a Poisson surface reconstruction [Kazhdan et al., 2006]. The output is a triangular mesh, each vertex consisting of a 3D point plus surface normal. This mesh is then refined as described in Section 5.2.2.

5.2 Refinement

This section describes the refinement method that was utilized in Section 5.1. The refinement is a linear combination of two terms: a photometric consistency term d_p that favors solutions with high NCC and a surface consistency term d_s that favors smooth solutions. These terms are balanced both by a user-specified smoothness parameter w_s and a data-driven parameter w_p, which ensures that the photometric term has greatest weight in regions with good feature localization. The refinement is performed both on the disparity map and later on the surface and we will discuss the individual realizations in Sections 5.2.1 and 5.2.2, resp. Both refinements are implemented as iterative processes. In practice they were found to preserve the volume and to converge quickly to the desired solution. Figure 5.4 shows the convergence for the disparity refinement. Since the convergence is close to exponential at the beginning, we terminate the refinement before convergence is reached to

strike a balance between quality and computational effort. This is especially valuable for lower-resolution layers of the disparity pyramid, since the next higher layer is going to refine the disparities anyway and we therefore need only to eliminate the gross errors.

5.2.1 Disparity Map Refinement

Sub-pixel disparity values are updated in every iteration as a linear combination of d_p and d_s, where d_p is an adjustment in the direction of improved photometric-consistency, and d_s is an adjustment in the direction of improved surface-consistency. Individual steps are -

Compute d_p Given current pixel p in image I and its match q in image J, compute the $\overline{\text{NCC}}$ of p with $q-1$, q, $q+1$ where the offsets indicate the left- and right-neighbors of q. We use $\overline{\text{NCC}} = (1 - \text{NCC})/2$, which resembles an error function ranging from 0 (no error) to 1 (complete dissimilarity). The respective $\overline{\text{NCC}}$s are labeled ξ_{-1}, ξ_0, ξ_{+1} and d_p is calculated as

$$d_p = \begin{cases} p - q - 0.5 & \xi_{-1} < \xi_0, \xi_{+1} \\ p - q + 0.5\frac{(\xi_{-1} - \xi_{+1})}{\xi_{-1} + \xi_{+1} - 2\xi_0} & \xi_0 < \xi_{-1}, \xi_{+1} \\ p - q + 0.5 & \xi_{+1} < \xi_{-1}, \xi_0 \end{cases} \tag{5.2}$$

Compute d_s The formulation of surface-consistency has been designed for human faces, where disparity varies smoothly with just a few (extreme) depth discontinuities. These discontinuities suggest the use of anisotropic kernels [Robert and Deriche, 1996], which adapt to the local gradient to avoid smoothening across boundaries. For human faces however, regions of high gradient are mostly due to different foreshortening of the camera pairs and smoothening should not be attenuated within these regions. Following [Woodford et al., 2008] we employ second-order properties, but use them within an anisotropic formulation over a two dimensional domain. The equation is discretized as

$$d_s = \frac{w_x(d_{x-1,y} + d_{x+1,y}) + w_y(d_{x,y-1} + d_{x,y+1})}{2(w_x + w_y)}, \tag{5.3}$$

where $w_x = \exp(-\left(\left|d_{x-1,y} - d_{x,y}\right| - \left|d_{x+1,y} - d_{x,y}\right|\right)^2)$. These weights render the harmonic equation anisotropic, reducing smoothing across depth discontinuities.

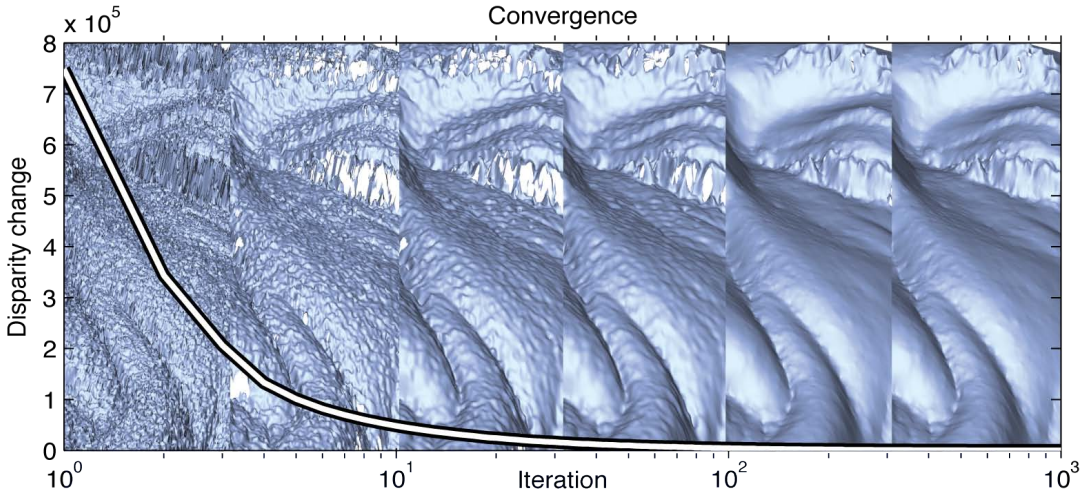


Figure 5.4: Convergence behavior - *Convergence of the refinement over the first 1000 iterations at layer ℓ_1 using $w_s = 0.005$. The images in the back show samples of the surface at iterations 0,1,5,10,100 and 1000, resp. The initial convergence is close to exponential (note the log scale for the iteration axis) and as can be seen from the samples the quality does not change noticeably between iterations 100 and 1000. We thus stop the refinement at iteration 180.*

Compute d' The equation is $d' = \left(w_p d_p + w_s d_s\right) / \left(w_p + w_s\right)$, where w_s is a user-specified smoothness parameter and w_p is

$$w_p = \begin{cases} \xi_0 - \xi_{-1} & \xi_{-1} < \xi_0, \xi_{+1} \\ 0.5(\xi_{-1} + \xi_{+1} - 2\xi_0) & \xi_0 < \xi_{-1}, \xi_{+1} \\ \xi_0 - \xi_{+1} & \xi_{+1} < \xi_{-1}, \xi_0 \end{cases} \tag{5.4}$$

Thus the photometric term has greatest weight in textured areas of the image where the image data is most informative about feature localization.

The refinement is terminated after a predefined number of iterations (40 for the lower-resolution layers and 180 for highest layer). See Figure 5.4 for justification.

5.2.2 Surface Refinement

The surface refinement differs from the disparity map refinement in that we need to refine in continuous 3-space. To keep computation tractable we restrict the refinement to along the normal direction $\mathbf{n}$ at $\mathbf{x}$ and define a refine-

ment resolution δ (usually 0.1 mm). The normals are not changed during the refinement. This again results in a discrete one-dimensional refinement and we proceed analogously to Section 5.2.1 by iterating over all points and computing updates for $\mathbf{x}$ as a linear combination of $\mathbf{x}_p$ and $\mathbf{x}_s$, where $\mathbf{x}_p$ is an adjustment in the direction of improved photometric-consistency, and $\mathbf{x}_s$ is an adjustment in the direction of improved surface-consistency. Individual steps are -

Compute $\mathbf{x}_p$ Generate the points $\mathbf{x}_{-\delta} = \mathbf{x} - \delta\mathbf{n}$ and $\mathbf{x}_{+\delta} = \mathbf{x} + \delta\mathbf{n}$. Define as reference view the visible camera with the least foreshortened view of $\mathbf{x}$. Measure a photo-consistency error for a point by taking the $\overline{\text{NCC}}$ between a 3x3 patch centered at the projection in the reference image and the corresponding patches in all other images where the patch is visible. Compute δ_p analogously to d_p given error values $\xi_{-\delta}, \xi_0$ and $\xi_{+\delta}$ for $\mathbf{x}_{-\delta}$, $\mathbf{x}_0$ and $\mathbf{x}_{+\delta}$, resp.

Compute $\mathbf{x}_s$ The surface-consistency estimate $\mathbf{x}_s$ is computed using mean-curvature flow [Meyer et al., 2003].

Compute $\mathbf{x}'$ Compute $\mathbf{x}' = (w_p\mathbf{x}_p + w_s\mathbf{x}_s)/(w_p + w_s)$ where w_p and w_s are the same as in Section 5.2.1.

5.2.3 Modeling Mesoscopic Geometry

The refinement in Section 5.2.2 results in surface geometry that is smooth across skin pores and fine wrinkles, because the disparity change across such a feature is too small to detect[3]. The result is flatness and lack of realism in synthesized views of the face. On the other hand, visual inspection shows the obvious presence of pores and fine wrinkles in the images. This is due to the fact that light reflected by a diffuse surface is related to the integral of the incoming light. In small concavities, such as pores, part of the incoming light is blocked and the point thus appears darker. This fact has been exploited by various authors (e.g. [Glencross et al., 2008]) to infer local geometry variation. In this section we propose a method to embed this observation into our surface refinement framework. It is qualitative, and the geometry that is recovered is not metrically correct. However, modulation of the macroscopic geometry with fine-scale features does produce a significant improvement in the perceived quality of reconstructed face geometry.

[3] This is a function of image resolution, not a limitation of the algorithm.

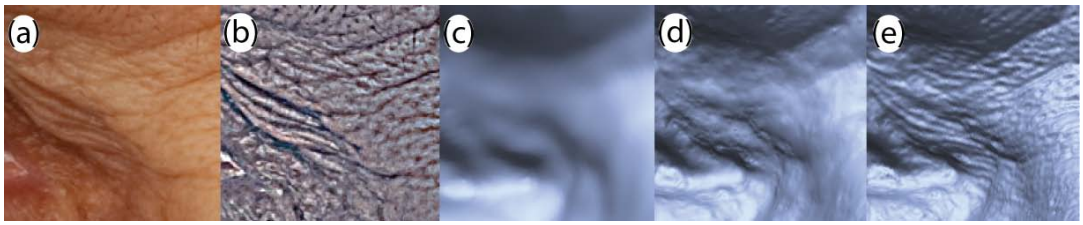


Figure 5.5: Mesoscopic augmentation - *This figure demonstrates the effect of the mesoscopic-consistency term. The captured image (a) is filtered to extract the mesoscopic detail (b). In (c), the Poisson-reconstructed surface is shown. The refinement described in Section 5.2.2 already enhances the coarse geometry (d), but only the mesoscopic formulation is capable of reproducing the fine-scale details (e).*

5.2.3.1 Computing Mesoscopic Values

For the mesoscopic augmentation we are only interested in features that are too small to be recovered by the stereo algorithm while still being visible in the captured images. We call these features mesoscopic features. A first step is to extract the mesoscopic features from the images using a high-pass filter. The filter is implemented as a Difference of Gaussians (DoG) using the original and a low-pass filtered version of the image. The applied low-pass filter is a Gaussian $\mathcal{N}(0, \sigma)$. This Gaussian is defined on the surface and projected into the views to account for perspective [Zwicker et al., 2002]. The standard deviation σ can either be defined directly on the surface knowing the size of the mesoscopic features or in the image domain and then propagated to the surface. Defining σ based on the image domain is more general, since the mesoscopic scale is relative to the projected size of the features in the image domain. The following derivation of σ is thus given in the image domain but its application to the surface is straight-forward.

If a feature covers only a few pixels, potentially less than the matching window, then the stereo reconstruction cannot match parts of the feature to reconstruct it. We thus choose the lowest spatial frequency of a feature (e.g. pore) that exactly covers the matching window m as the cutting frequency $f_{cut} = 1/2m$. Higher spatial frequencies should be retained while the lower ones should be (decreasingly) attenuated. Since we use a Gaussian as low-pass filter, this is the case when the cutting frequency equals $f_{cut} = n\sigma_F$ with $n \approx 1.5$. The variance in the frequency domain is then given as $\sigma_F^2 = (f_{cut}/n)^2$ and can be converted to the variance σ_f^2 in the spatial domain using the un-

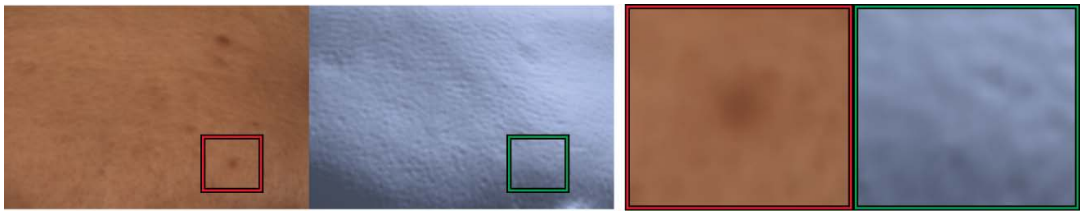


Figure 5.6: Frequency behavior - *This figure shows part of a forehead on the left and a zoomed-in patch on the right. Note that the mesoscopic augmentation adds high-frequency details such as pores, while coarser features, such as the spot showed on the right, usually do not influence the geometry, since they usually do not contain very high spatial frequencies.*

certainty principle for Gaussians

$$\sigma_f^2 = \frac{1}{4\sigma_F^2} \tag{5.5}$$

and thus

$$\sigma_f = nm. \tag{5.6}$$

Given the proposed values $n = 1.5$ and $m = 3$, the standard deviation is $\sigma_f \approx 4.5$ pixels. These computations are performed for the camera in which the projected features cover the least pixels (usually the camera with lowest resolution) and propagated to the surface of the subject as σ.

Using the projection of the previously defined Gaussian $\mathcal{N}(0, \sigma)$, we compute mesoscopic values μ for all points $\mathbf{x}$

$$\mu(\mathbf{x}) = \frac{\sum_{c \in \mathcal{V}} \alpha_c \left(I_c(P_c(\mathbf{x})) - \left[\mathcal{N}_{\mathbf{\Sigma}_c} * I_c \right] (P_c(\mathbf{x})) \right)}{\sum_{c \in \mathcal{V}} \alpha_c}, \tag{5.7}$$

where $\mathcal{V}$ denotes the set of visible cameras, $P_c(\mathbf{x})$ the projection of $\mathbf{x}$ into camera c, $\mathbf{\Sigma}_c$ the covariance matrix of the projection of the Gaussian $\mathcal{N}(0, \sigma)$ into camera c, and the weighting term α_c is the cosine of the foreshortening angle observed at camera c.

5.2.3.2 Mesoscopic Augmentation

The next steps are based on the assumption that variation in mesoscopic intensity is linked to variation of the geometry. For human skin we found that

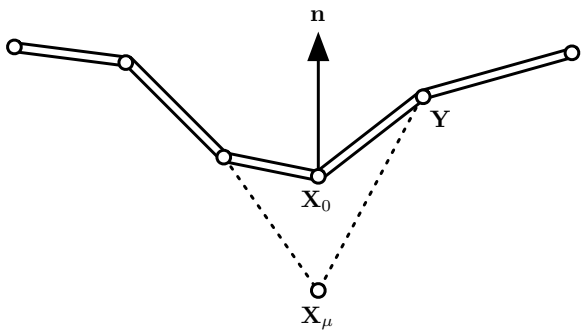


Figure 5.7: Local geometry - *The solid line represents the current surface estimate while the dotted line indicates the true surface. The local geometry $G_{\mathbf{x}_\mu}$ for a point $\mathbf{x}_\mu$ is defined as $\langle \mathbf{y} - \mathbf{x}_\mu, \mathbf{n} \rangle$ for all points in the neighborhood. This also includes the point itself $G_{\mathbf{x}_\mu}(\mathbf{x}_\mu) = 0$ and the current estimate $G_{\mathbf{x}_\mu}(\mathbf{x}_0) = \delta$ given that $\mathbf{x}_\mu = \mathbf{x}_0 + \delta\mathbf{n}$. Note that $\mathbf{x}_\mu$ is the desired position, while $\mathbf{x}_0$ is its current estimate.*

this is mostly the case. Spatially bigger skin features tend to be smooth and are thus filtered out as shown in Figure 5.6. The idea is thus to adapt the local high-frequency geometry of the mesh to the mesoscopic field $\mu(\mathbf{y})$ for a point $\mathbf{y}$ on the surface. The geometry should locally form a concavity whenever $\mu(\mathbf{y})$ decreases and a convexity when it increases. We can thus write $\frac{d}{d\mathbf{y}} G_{\mathbf{x}_\mu}(\mathbf{y}) = \frac{d}{d\mathbf{y}} f_\mu(\mu(\mathbf{y}))$ where f_μ is a function that relates mesoscopic variation to geometric variation. The local geometry $G_{\mathbf{x}_\mu}(\mathbf{y})$ for a patch shown in Figure 5.7 is expressed relative to the point $\mathbf{x}_\mu$ as $\langle \mathbf{y} - \mathbf{x}_\mu, \mathbf{n} \rangle$ for a point $\mathbf{y}$ in the neighborhood of $\mathbf{x}_\mu$, which denotes the desired estimate of $\mathbf{x}$. Given the current estimate $\mathbf{x}_0$ we can write $\mathbf{x}_\mu = \mathbf{x}_0 + \delta\mathbf{n}$.

Using finite differences to approximate the differentials leads to

$$\frac{G_{\mathbf{x}_\mu}(\mathbf{y}) - G_{\mathbf{x}_\mu}(\mathbf{x}_\mu)}{\|\mathbf{y} - \mathbf{x}_\mu\|} = \frac{f_\mu(\mu(\mathbf{y})) - f_\mu(\mu(\mathbf{x}_\mu))}{\|\mathbf{y} - \mathbf{x}_\mu\|}. \tag{5.8}$$

After expanding $G_{\mathbf{x}_\mu}$ and simplifying the equation becomes

$$\langle \mathbf{y} - \mathbf{x}_\mu, \mathbf{n} \rangle - \langle \mathbf{x}_\mu - \mathbf{x}_\mu, \mathbf{n} \rangle = f_\mu(\mu(\mathbf{y})) - f_\mu(\mu(\mathbf{x}_\mu)). \tag{5.9}$$

As f_μ we choose the multiplication with a user defined parameter η — the embossing strength. Using the fact that $\mathbf{x}_\mu = \mathbf{x}_0 + \delta\mathbf{n}$ and the assumption that the mesoscopic field remains constant along $\mathbf{n}$, we find

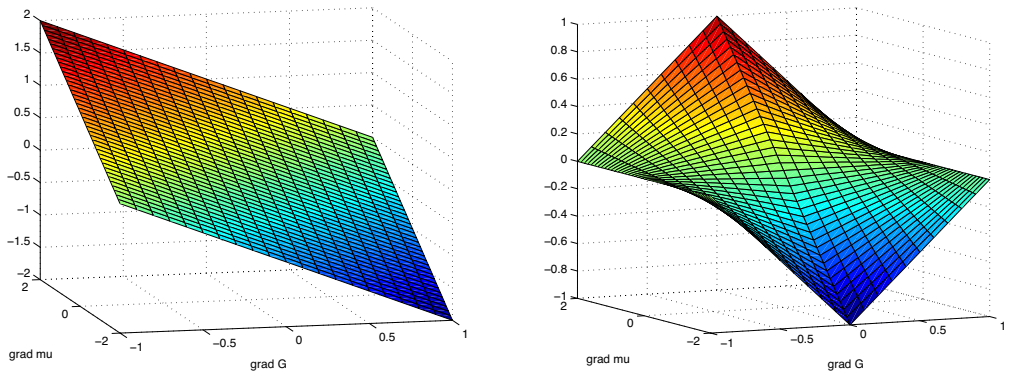


Figure 5.8: Comparison of the correctional factor δ *- Analytic derivation (left) and the empirically modified version (right). The modified version has the property that the correctional factor* δ *is attenuated when there is no or little high-frequency content in the image and it is even further attenuated whenever the geometric gradient is large. This reduces the impact of the mesoscopic term on high-frequency features that can be reconstructed by the MVS, such as hair. On the other hand, the correctional factor reaches its maximum when the mesoscopic gradient is large and the geometric gradient small — augmenting flat surfaces with high-frequency detail.*

$$\langle \mathbf{y} - \mathbf{x}_0 - \delta \mathbf{n}, \mathbf{n} \rangle = \eta \left(\mu(\mathbf{y}) - \mu(\mathbf{x}_0) \right), \tag{5.10}$$

which can be rearranged to

$$\delta = \eta \left(\mu(\mathbf{x}_0) - \mu(\mathbf{y}) \right) - \langle \mathbf{x}_0 - \mathbf{y}, \mathbf{n} \rangle . \tag{5.11}$$

Intuitively, this function states that the point needs to change along the normal proportional to the difference of the mesoscopic and geometric gradients. Since we are using the full geometry on the one hand but only the high-frequency content of the shading on the other, this function smoothens the mesh. To avoid this effect, we empirically modify equation 5.11 to become

$$\delta = \eta \left(\mu(\mathbf{x}_0) - \mu(\mathbf{y}) \right) \left(1 - \frac{|\langle \mathbf{x}_0 - \mathbf{y}, \mathbf{n} \rangle|}{\|\mathbf{x}_0 - \mathbf{y}\|} \right). \tag{5.12}$$

This equation has the property that the correctional factor δ is attenuated when there is no or little high-frequency content in the image and it is even

Algorithm 1: - $\mathbf{x}' = refinePointMesoscopic(\mathbf{x}, \mathbf{n}, \delta, w_s, \rho, \eta)$
- NCC(**x**,**n**) computes the normalized cross correlation of a surface patch at **x** with normal **n** by projecting it into all visible images
- $\bar{\kappa}$ denotes the mean curvature

$\xi_{-\delta} = (1 - \mathrm{NCC}(\mathbf{x} - \delta\mathbf{n}, \mathbf{n}))/2$
$\xi_0 = (1 - \mathrm{NCC}(\mathbf{x}, \mathbf{n}))/2$
$\xi_{+\delta} = (1 - \mathrm{NCC}(\mathbf{x} + \delta\mathbf{n}, \mathbf{n}))/2$
if $\xi_{-\delta} < \xi_{+\delta}, \xi_0$ **then**
  $\delta_p = -0.5\delta$
  $w_p = (\xi_0 - \xi_{-\delta})/\delta$
else if $\xi_{+\delta} < \xi_{-\delta}, \xi_0$ **then**
  $\delta_p = 0.5\delta$
  $w_p = (\xi_0 - \xi_{+\delta})/\delta$
else
  $\delta_p = 0.5(\xi_{-\delta} - \xi_{+\delta})/(\xi_{-\delta} + \xi_{+\delta} - 2\xi_0)\delta$
  $w_p = 0.5(\xi_{-\delta} + \xi_{+\delta} - 2\xi_0)/\delta$
end if
$\delta_s = -\bar{\kappa}\mathbf{n}$
$\delta_\mu = \eta \frac{\sum_{i\in\mathcal{R}} \exp(-\|\mathbf{x}-\mathbf{x}_i\|)(\mu(\mathbf{x})-\mu(\mathbf{x}_i))(1-|\langle \mathbf{x}-\mathbf{x}_i, \mathbf{n}\rangle|/\|\mathbf{x}-\mathbf{x}_i\|)}{\sum_{i\in\mathcal{R}} \exp(-\|\mathbf{x}-\mathbf{x}_i\|)}$
$w_\mu = 3\rho\xi_0/\delta(\xi_{-\delta} + \xi_0 + \xi_{+\delta})$
$\mathbf{x}' = \mathbf{x} + (w_p\delta_p + w_s\delta_s + w_\mu\delta_\mu)/(w_p + w_s + w_\mu)\mathbf{n}$

further attenuated whenever the geometric gradient is large (see Figure 5.8). This reduces the impact of the mesoscopic term on high-frequency features that can be reconstructed by the MVS, such as hair. On the other hand, the correctional factor reaches its maximum when the mesoscopic gradient is large and the geometric gradient small — augmenting flat surfaces with high-frequency detail. An alternative approach would be to use a high-pass filter on the geometry as well. This however increases computational cost substantially, since the filter needs to be applied at every iteration step, and high-frequency detail reconstructed by the MVS would be submersed.

The correctional factors are then integrated over a region of support $\mathcal{R}$ (the 1-ring in our implementation) as

$$\delta_\mu = \frac{\sum_{i\in\mathcal{R}} w_i\delta_i}{\sum_{i\in\mathcal{R}} w_i}, \tag{5.13}$$

where the weights are computed by a radial basis function such as $w_i = \exp(-\|\mathbf{x} - \mathbf{x}_i\|)$.

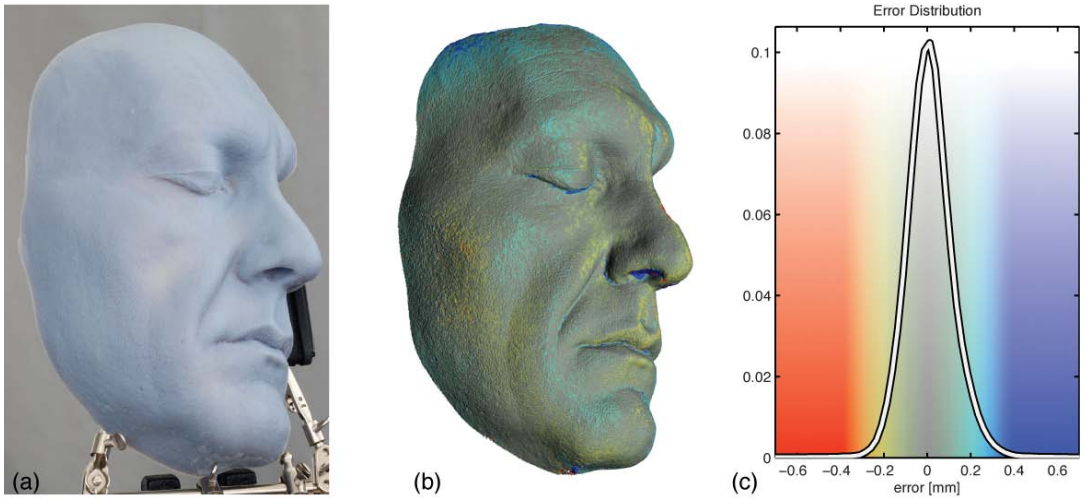


Figure 5.9: Quantitative comparison - *(a) Physical mask of known ground-truth; (b) Recovered model color-coded by error; (c) Distribution of the signed absolute error between the ground-truth and the registered recovered model.*

The update of the 3D point now uses all three adjusted points $\mathbf{x}_p$, $\mathbf{x}_s$ and $\mathbf{x}_\mu$ to compute $\mathbf{x}' = (w_p\mathbf{x}_p + w_s\mathbf{x}_s + w_\mu\mathbf{x}_\mu)/(w_p + w_s + w_\mu)$. The weights w_p and w_s are the same as in Section 5.2.2 and w_μ is defined as

$$w_\mu = \rho\frac{3\xi_0}{\delta(\xi_{-\delta} + \xi_0 + \xi_{+\delta})}, \tag{5.14}$$

with ρ being a user specified term that controls the influence of the mesoscopic term. Figure 5.5 shows an example of how the mesoscopic term enriches the results. To emphasize the simplicity of the refinement we provide pseudocode in Algorithm 1. The algorithms for the refinements described in Sections 5.2.1 and 5.2.2 are very similar and of less complexity, since the mesoscopic term is not present. The function computes the position update $\mathbf{x}'$ for $\mathbf{x}$ using the normal $\mathbf{n}$. The parameters and their typical values are: resolution $\delta = 0.05$ in mm, surface smoothness $w_s = 0.03$, mesoscopic weight $\rho = 0.07$ and embossing strength $\eta = 0.2$.

5.2.4 Variation of the Refinement

In this section we describe a variation of the refinement step that was introduced for Chapter 7. All results in this chapter were produced with the

	Average [mm]	Median [mm]	Angular [°]
PMVS	0.132 ± 0.19	0.096	6.989 ± 7.97
Ours w/o Mesoscopic Term	0.092 ± 0.13	0.070	8.690 ± 7.69
Ours with Mesoscopic Term	$\mathbf{0.088 \pm 0.12}$	**0.067**	$\mathbf{5.675 \pm 6.03}$

Table 5.1: *Absolute error values for the ground-truth experiment described in Section 5.3.1. The angular error measures the angular difference of the normals. The refinement with mesoscopic term is superior in all cases. We included a comparison to PMVS [Furukawa and Ponce, 2007] for completeness, however we want to point out that their method is a general-purpose MVS, while ours is tailored to face-reconstruction.*

initial refinement formulation. The original refinement strategy produces very noisy surfaces in areas which are not continuous, diffuse and smooth surfaces. An example for such an area is hair. For the performance capture algorithm introduced in Chapter 7, it is preferable to have smooth surfaces in these areas, as noisy surfaces will cause temporal flickering. For this, the smoothness weight w_s introduced in Section 5.2.2, is changed to

$$w_s' = \lambda_0^s + \lambda_1^s \xi_0 + \lambda_2^s \xi_0^2, \tag{5.15}$$

where ξ_0 is the matching error and λ^s is a user defined smoothness coefficient vector. Note that the original smoothness weight is a special case of this variant with $\lambda^s = (w_s, 0, 0)$.

5.3 Results

5.3.1 Quantitative Evaluation

This section contains results for a physical mask of known ground-truth. The mask was created by taking a plaster-cast of a face, scanning with laser, and printing on an Object Connex 500 3D printer. Figure 5.9 shows the mask which is half a face, not a full face, due to an unwanted limitation at the time of our experiments. Error is measured as perpendicular distance between the registered ground-truth model and recovered model. The errors are listed in Table 5.1 and their distribution is shown in Figure 5.9. For comparison, the physical resolution of the 3D printer is 0.042 mm. The error statistics include regions like the nostrils (scale $\sim$ 5 mm), where the algorithm did not reconstruct because the nostril interior is invisible in the images. Thus, the errors

Figure 5.10: Qualitative comparison - *Top: rendering of the physical mask; Center: image of the physical mask; Bottom: rendering of the recovered model.*

are an over-estimate in the sense that they include this source, but removal would have invalidated the objectivity of the result. Figure 5.10 provides a visual comparison of the ground-truth and recovered models. The details in the recovered model are slightly less defined but recovery of mesoscopic geometry is substantially correct.

Measured errors are not directly applicable to real faces because the surface reflectance of the face mask is different from human skin, with reduced specularity for example, but the results are informative to first-order. These ex-

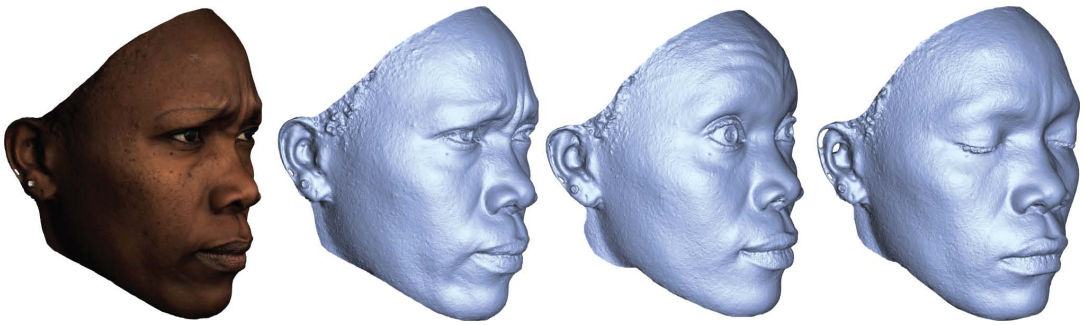


Figure 5.11: *Recovered model for a face with dark-colored skin.*

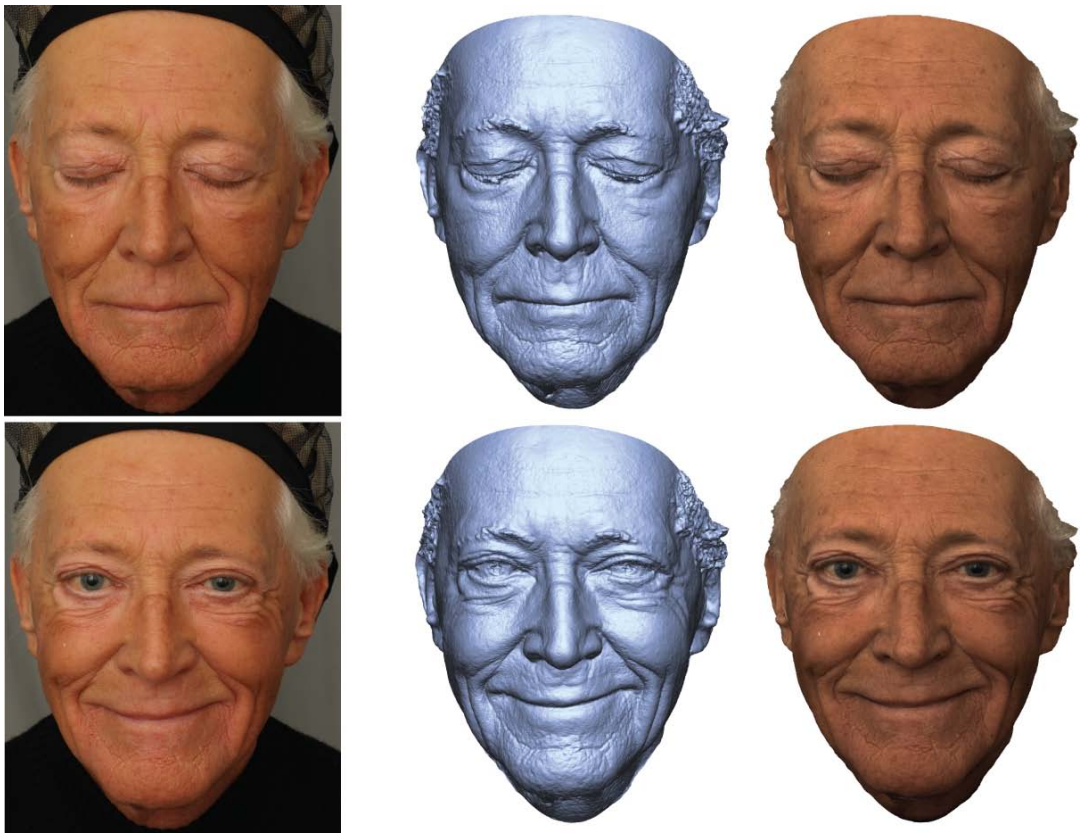


Figure 5.12: *Recovered models of a subject for two different expressions.*

periments have also suggested an interesting possibility for future work — latest-generation 3D printers have sophisticated material handling, and it might be possible to print a mask whose reflectance properties are a better approximation to skin.

5.3.2 Qualitative Evaluation

Figure 5.13 shows results for a variety of subjects of varying gender, ethnicity, age, and facial expression. Figure 5.12 demonstrates a high-fidelity reconstruction for an elderly male model. Figure 5.12 shows both the subtle deformations of mesoscopic detail in distorted areas as well as their consistency in regions that do not undergo deformation. Figure 5.11 shows results for a subject with dark-colored skin. Figures 5.18 and 5.19 demonstrate high-fidelity reconstruction for an elderly female model without and with texture, resp.

Figure 5.13: *Recovered models and synthesized views, for viewpoints different from the original camera images, across subjects of varying appearance. Our focus has been on skin, and it may be that the hair and specular components — like the eyes, teeth and tongue — benefit from custom algorithms. But 3D reconstruction is reasonable across all these parts.*

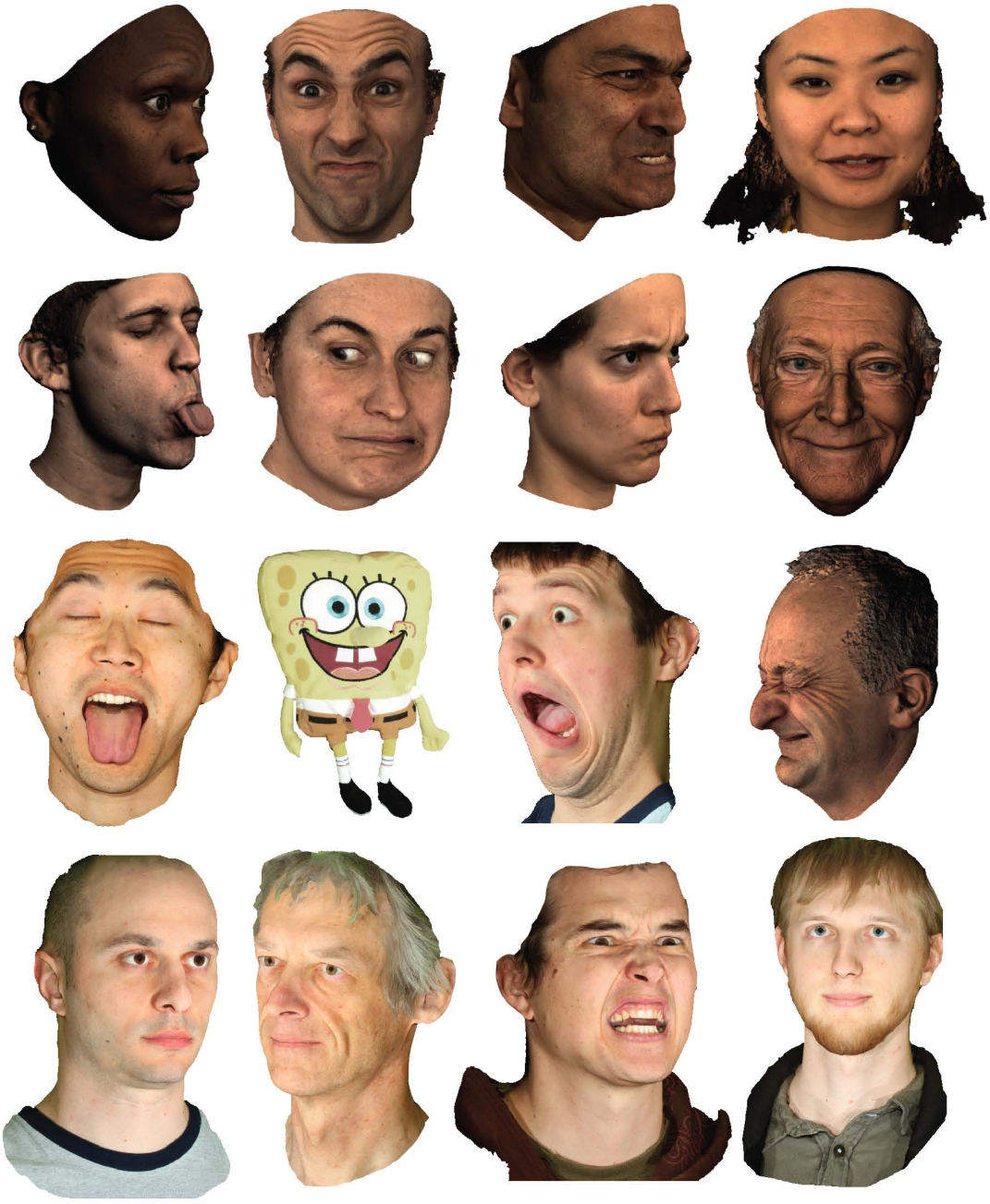


Figure 5.14: *The scans shown in Figure 5.13 with applied texture. Our focus has been on skin, and it may be that the hair and specular components — like the eyes, teeth and tongue — benefit from custom algorithms. But 3D reconstruction is reasonable across all these parts.*

Figure 5.15: *Top: Images of a subject slapping himself and causing a shock-wave in the face. Bottom: the respective reconstructions.*

Figure 5.15 shows models recovered for highly-transient facial expression. The subject slapped his own cheek causing a fast-moving shock-wave across the face. The results illustrate the advantage of single-shot capture — a time-multiplexed system would require specialized high-speed hardware and high light-levels for this case.

Figure 5.16 shows results for capture from the Fuji camera. Image capture with the Fuji under normal ambient light yielded very noisy images, most likely due to the relatively small 1/2.3 inch sensor size. Doing the capture with a bright diffuse light source solved this problem and yielded the required image quality. The face model has less coverage than with the studio setup, because this is a small baseline stereo camera taking a frontal view.

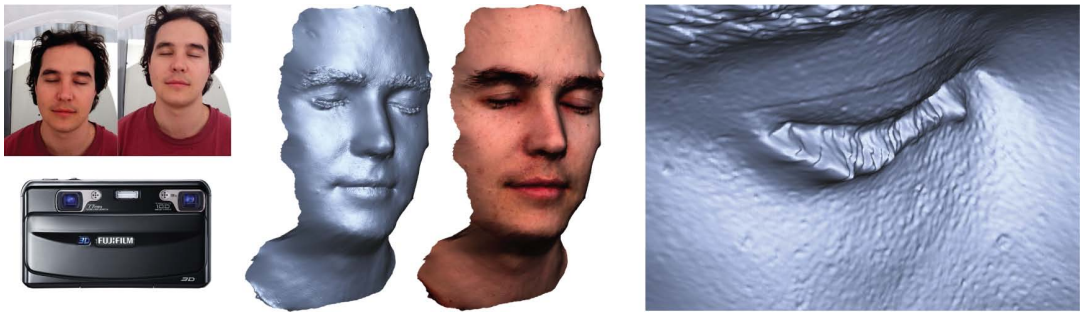


Figure 5.16: *Left: Images from the Fuji binocular-stereo camera. Center: the recovered model. Right: Close-up of a region around the eye.*

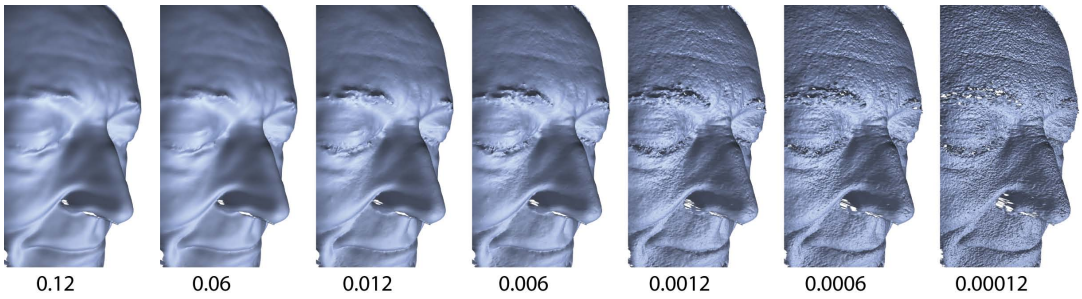


Figure 5.17: Smoothness parameter - *Influence of the smoothness parameter w_s on the disparity map. The figure shows the results after 180 iterations. The parameter is stable and well behaved.*

5.3.3 Robustness

The presented system has only very few parameters that require user adjustment. The most important ones are the smoothness parameter w_s (Sections 5.2.1 and 5.2.2), the mesoscopic influence ρ (Section 5.2.3) as well as the mesoscopic embossing strength η (Section 5.2.3). This is an important property of the system, since there is little or no hand-tuning required for individual subjects. All subjects in this chapter have been computed using the same parameters, except for subjects with darker skin where the smoothness parameter w_s was increased slightly to adjust for the lower signal-to-noise ratio. Furthermore, the parameters are stable since a small change in parameter space leads to a small change in the resulting solution as demonstrated in Figure 5.17 for the smoothness parameter.

5.3.4 Performance

The compute time, from image input to output of a 3D model, depends on several different factors; the amount of cameras, the desired mesh resolution, the amount of Bundle Adjustment steps, and of course processing power. Reconstructing a facial scan at highest resolution with our eight-camera DSLR setup using three steps of Bundle Adjustment requires approximately 20 minutes on a desktop computer (Intel Xeon 3.33GHz). On the same computer, reconstructing scans captured with our seven-camera performance capture setup without Bundle Adjustment takes slightly over one minute per frame when pipelining several frames. A related matter of practical usefulness is that the stereo matching is pyramidal and it is straightforward to quickly generate models at the lower-resolution layers for preview

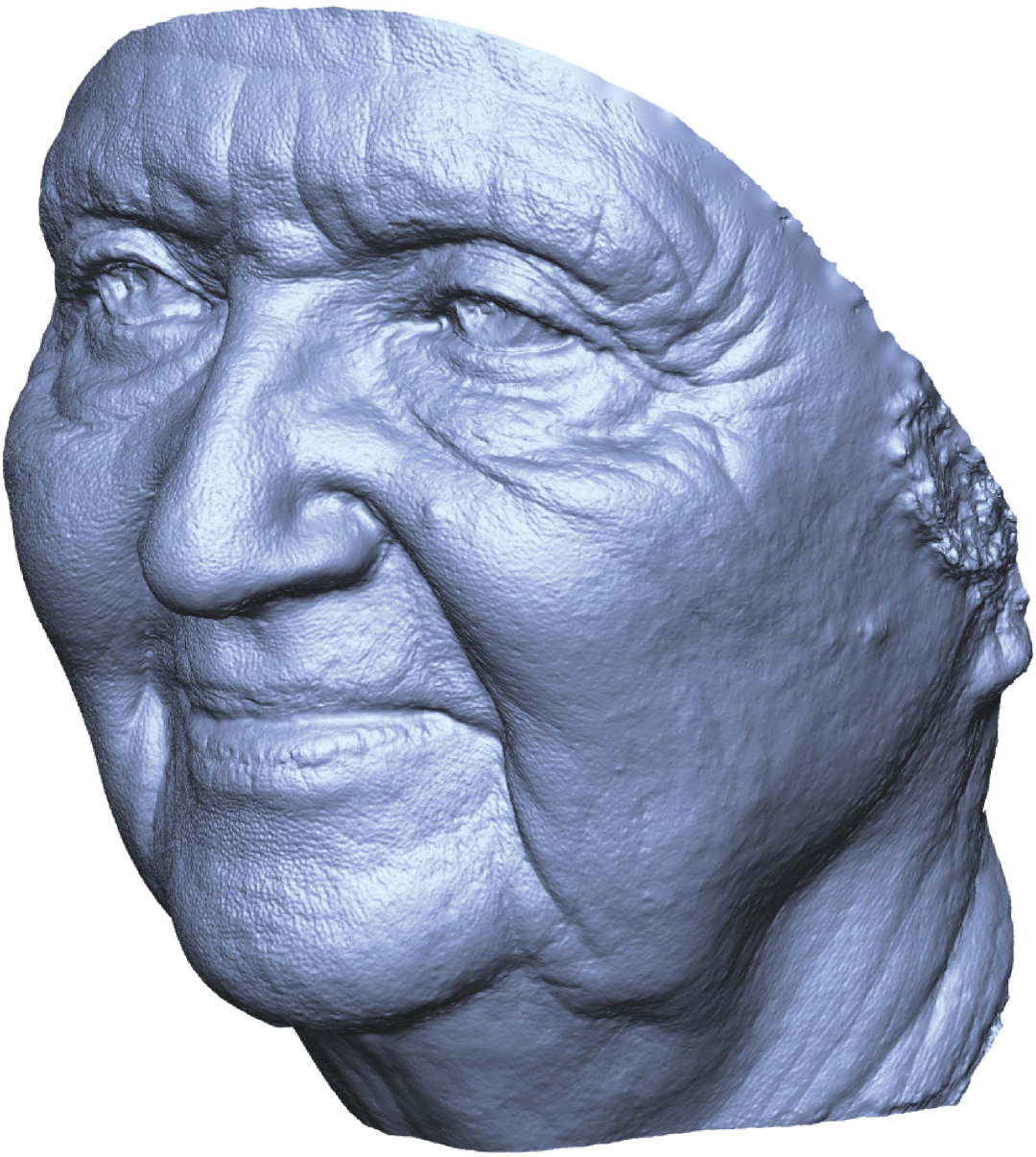


Figure 5.18: *High resolution reconstruction of an older female model. Note the geometry of the eye. The cornea of the eye is transparent and thus not reconstructed. The iris lying behind it on the other hand is opaque and can therefore be reconstructed.*

Figure 5.19: *The same scan as shown in Figure 5.18 but with applied texture. Texture is recovered from the same images that are used for reconstruction and is thus inherently compatible with the scan. No additional alignment is required for passive MVS systems.*

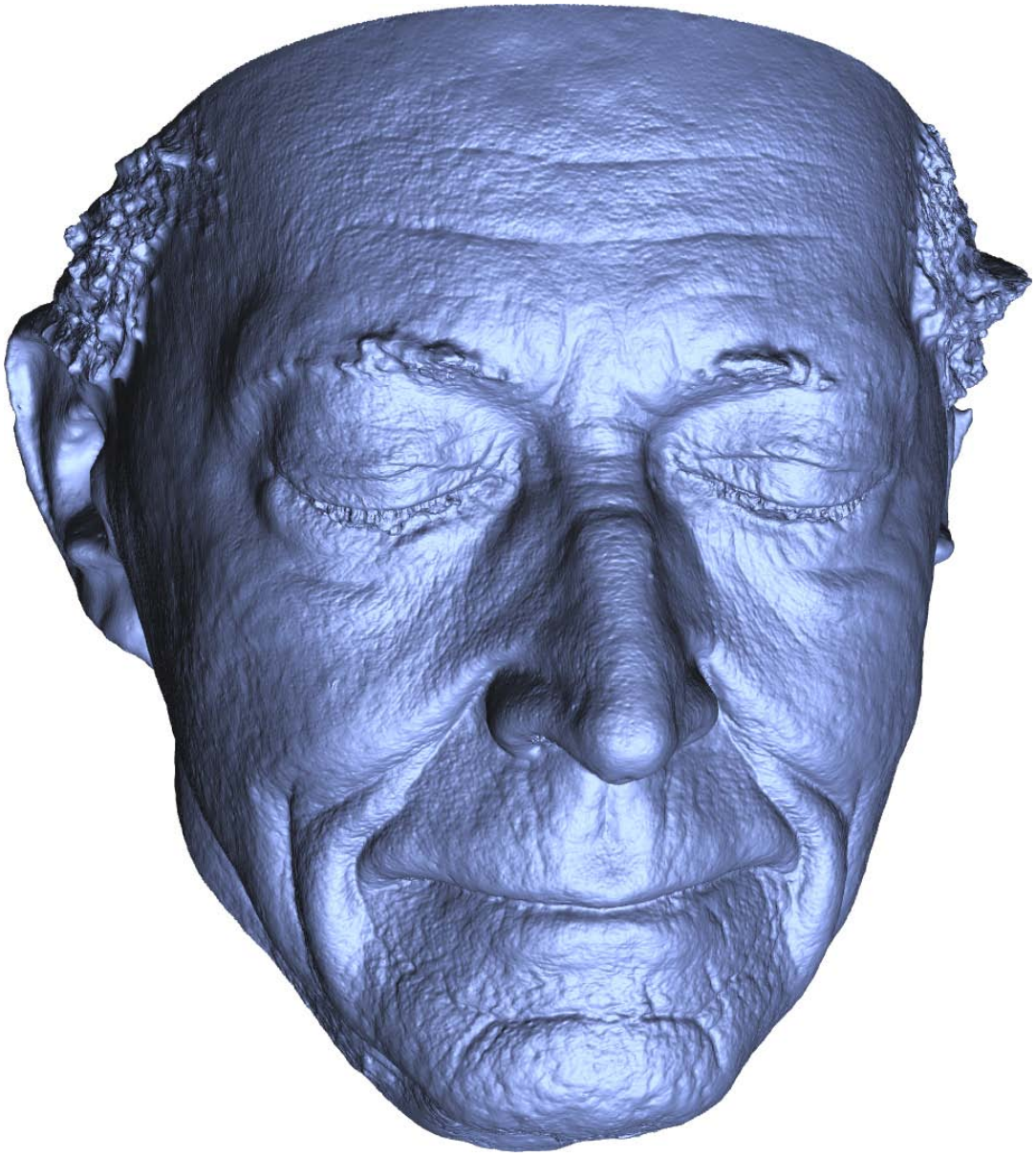


Figure 5.20: *High resolution reconstruction of an older male model.*

Figure 5.21: *The same scan as shown in Figure 5.20 but with applied texture. Texture is recovered from the same images that are used for reconstruction and is thus inherently compatible with the scan. No additional alignment is required for passive MVS systems.*

and checking. Model generation takes a few seconds at the lowest-resolution (150x150 pixel) layer for example.

5.4 Discussion

Robustness As discussed in Chapter 3 we used several contrasting capture systems, from a single consumer binocular-stereo camera, over setups with various numbers of prosumer DSLR camera (4-14) to machine vision video cameras. This illustrates system behavior on a spectrum ranging from careful capture of high-quality images to point-and-shoot capture of lower-quality images with lens distortion. It further illustrates system behavior on the spectrum of varying camera configuration, ranging from cameras all around the front hemisphere of the head to binocular stereo with a small baseline. All capture methods yield good quality face models, providing evidence that our calibration method and run-time system are robust to changing camera configuration and changing image characteristics. We have built face models for several hundred different subjects at this stage and for some of the subjects we reconstructed several thousand scans — without a single failure case and without the need to carefully tune the software for individual cases.

Current Limitations The system is tailored to reconstruct continuous, opaque and diffuse surfaces that exhibit texture. If these assumptions are violated, reconstruction quality degrades. Specularity on the face for example is a problem when doing capture under direct lighting, occurring for example when the tip of the nose reflects a bright light source. Specular areas typically distort the mesh. Ways to deal with this include preventing it from happening in the first place by using indirect lighting or cross-polarization as discussed in Chapter 3, or post-processing to explicitly detect the affected area and create a plausible reconstruction. Other areas that violate these assumptions are eyes, teeth or hair. Even though the algorithm is not meant to reconstruct these features, it fails gracefully. The eyes for example are reconstructed without the transparent cornea, but otherwise correct and hair is reconstructed as a shrink-wrapped surface that covers the hair volume.

5.5 Conclusion and Future Work

The best current methods to obtain high-quality face models use active light, and they offer reliability and accuracy. For example, laser is noted for its ability to produce point measurements of sub-millimeter accuracy, while gradient-based illumination has the ability to accentuate detail and enhance recovery of fine-scale 3D geometry. However active methods impose constraints such as the need for special-purpose hardware, for subjects to be still, or for projected light that is intrusive due to high-brightness or strobing.

In contrast, passive stereo vision uses single-shot capture under standard light sources. And commodity cameras now routinely have the image resolution to reveal individual skin pores, so that faces provide the kind of dense evenly-distributed texture that is perfect for stereo matching and 3D reconstruction. This chapter has demonstrated the capabilities of a state-of-the-art passive stereo system for face scanning. It competes with active systems in reliability and quality for high-end applications, but it is low-cost, and versatile enough to work off a consumer stereo camera. We demonstrated an augmented type of stereo refinement to qualitatively recover pore-scale geometry and yield improved visual realism in synthesized faces.

The proposed system has been designed to capture continuos surfaces such as human skin. While the system still provides reasonable results in areas that do not comply with these assumptions, such as hair or eyes, we believe that dedicated reconstruction algorithms are required to provide the correct geometry. One such algorithm is described in Chapter 6 and permits coupled reconstruction of skin surface and sparse facial hair. Algorithms to reconstruct eyes or teeth for example are not in the scope of this thesis and offer an exciting are for future work.

The system requires only a single exposure, making it well suited to facilitate performance capture as shown in Chapter 7. In conclusion, we believe that this chapter demonstrates that passive stereo has matured into a robust technology for capturing models of the face, and that its advantages will support new types of deployment.

CHAPTER 6

Facial Hair

Since facial appearance plays such an important role in human communication, mastering the human face has long been a central goal of computer graphics. The characteristics of someone's face are a core component of their individuality and help make their physical appearance unique from every other person. While many facial characteristics are difficult to change, facial hair is one feature that is easily adapted. Some individuals meticulously sculpt their eyebrows hair-by-hair to ensure that the overall shape is perfectly formed and symmetric. A clean-shaven male face can look boyish and innocent. Many men instead choose to convey a more rugged, masculine appearance through a nearly unlimited variety of facial-hair styles, including beards, mustaches, and sideburns of all shapes and sizes. The popularity of these different facial-hair styles can fluctuate just as rapidly as fashion trends and varies dramatically from region to region, making facial hair a core piece of popular culture.

Despite the important role that facial hair plays in individual expression, existing face-capture technology does not easily accommodate facial-hair features. The problem setting is especially difficult because the hair and the underlying skin are often both visible, and a faithful face reconstruction must deliver accurate geometry for both the hair fibers and the underlying skin geometry. An accurate scan of a person with a scruffy beard should include

each visible beard hair together with a high-fidelity representation of the skin underneath. Capture algorithms, such as the one presented in Chapter 5, that focus on skin only will often deliver a skin surface that is "shrink-wrapped" around the facial-hair features, rather than reconstructing them as individual fibers on top of a skin surface. On the flip side, algorithms specialized for hair reconstruction typically focus on statistical properties of a thick head of hair that fully obscures the underlying scalp. These assumptions obviate the need for skin reconstruction or the consideration of individual fibers in isolation, making the algorithms unsuited for facial-hair reconstruction. As a result, existing scanning systems quietly ignore the huge variety of facial hair styles, treating them as error cases rather than as unique forms of human expression that deserve accurate reconstruction.

The research presented in this chapter improves the state-of-the-art of face capture with an algorithm that treats hair and skin surface capture together in a coupled fashion so that a high-quality representation of hair fibers as well as the underlying skin surface can be reconstructed. Since individual hair features are extremely fine and can vary greatly with head movement, we propose a single-shot capture system that uses consumer digital cameras without the need of multiple exposures or active illumination. Our hardware setup supports a variable number of cameras, so that additional face coverage is achieved simply by adding more cameras. All imagery needed for accurate reconstruction is captured within the time period of a single exposure.

Our reconstruction algorithm processes the captured images using a steerable filter kernel for explicit hair detection and produces a hair map for each image. Within these hair maps individual hairs are traced and then reconstructed and refined in 3D using multi-view stereo. We then employ a skin reconstruction algorithm that uses information about detected hair pixels and the reconstructed hair fibers to deliver a skin surface that lies underneath all hairs irrespective of hair occlusions. In sparse regions where individual hairs are clearly visible in the captured images, our algorithm reconstructs each and every detected hair as a collection of line segments. In dense regions, such as the eyebrows, where many hairs overlap and obscure one another, our algorithm employs a hair-synthesis method to create hair fibers that plausibly match the image data. Likewise, when skin is visible through sparse hair, our system accurately reconstructs it, and when skin is obscured our system proposes a plausible solution. We demonstrate this algorithm with a collection of scans of individuals exhibiting a variety of different facial-hair styles.

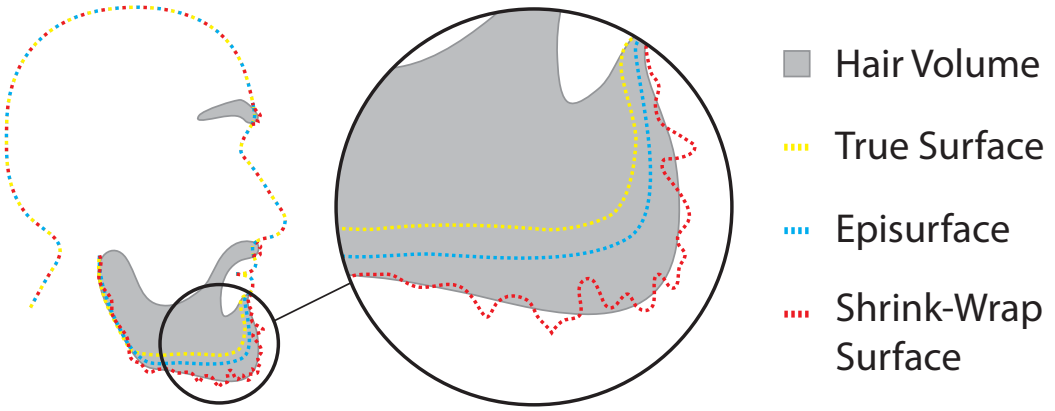


Figure 6.1: Skin episurface - *The skin episurface is a pseudo-surface that closely matches the real skin surface when the skin is clearly visible and passes below the visible hairs when the hair is dense.*

As a useful geometric construct for the system, we introduce the concept of the *skin episurface* (see Figure 6.1). In the case where there is no hair or low density hair over visible skin, the skin episurface is a close approximation to the true skin surface. In the case where the hair is dense and no skin is visible, the skin episurface is a postulated 3D surface below the top layer of visible hair. While it is not a true surface, the motivation for the construct is that it enables a unified approach to the processing, across areas of clear skin and dense beard. Note that it should not be thought of as a dilation of the skin surface — in the case of protruding facial hair, something like a goatee, the episurface will also form a protruding shape. In addition to the technical building blocks of our solution, we also show that the fidelity of face scans is enhanced when facial hair is accurately reconstructed and provides a more faithful representation of an individual's unique look.

6.1 Overview

This section contains a system overview for the 3D reconstruction of the skin episurface plus individual hairs. Many traditional 3D reconstruction algorithms treat the whole of a scene in an undifferentiated way. In contrast, the algorithm described here works by explicitly differentiating the skin episurface and the hair, and processing each of them in a distinct way.

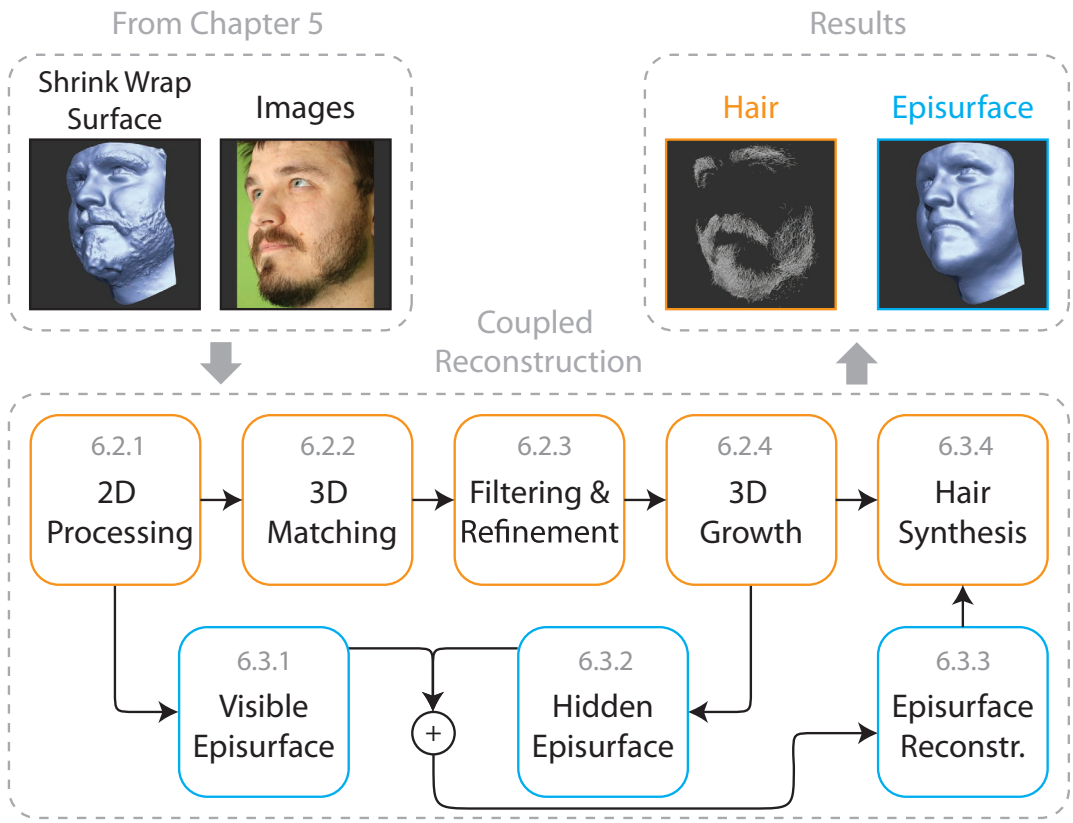


Figure 6.2: Main stages of the algorithm - *A preliminary step to the processing is to obtain a first estimate of the 3D model using the method presented in Chapter 5. The algorithm then explicitly detects and reconstructs hair fibers (top flow) and uses this information to provide a better estimate of the underlying surface (bottom flow). This surface, called the skin episurface, is in return used by the hair reconstruction to synthesize new hairs.*

The pipeline is:

(a) separate hairs and skin in the captured images, extract 2D hair fibers using a growing algorithm and remove the detected hairs from the images using inpainting;

(b) reconstruct, filter, refine and grow 3D fibers in 3-space based on the extracted 2D fibers using multi-view stereo (MVS);

(c) compute the skin episurface combining traditional MVS and the estimated roots of the 3D fibers;

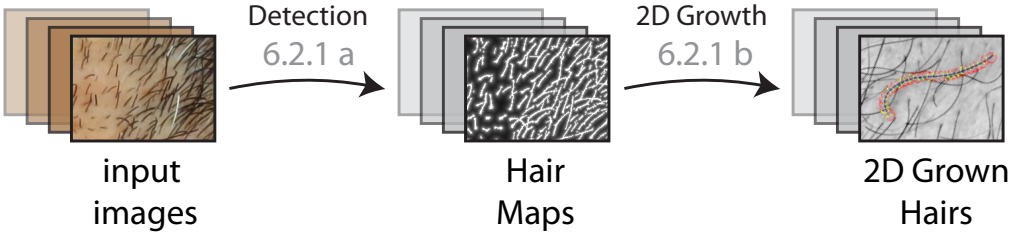


Figure 6.3: 2D processing - *Given a set of input images, each image is filtered in order to produce a hair map H (6.2.1a) used to grow 2D-hair fibers (6.2.1b).*

(d) synthesize hairs in areas where image data indicates the presence of hair, but individual hairs are indistinct and cannot be reconstructed.

Figure 6.2 illustrates the pipeline. Section 6.2 will describe stages (a) and (b), and Section 6.3 will describe stages (c) and (d).

A preliminary step to the processing below is to obtain a first estimate of the 3D model of the face using the method presented in Chapter 5, without any use of special algorithms for facial hair. We will use the terminology *shrink-wrap surface* below to refer to this first estimate of the 3D model, because it has the effect of wrapping the surface around the hairs. The shrink-wrap model will be used in Sections 6.2 and 6.3.

6.2 Computing 3D Hair

This section describes the computation of hair maps in the image, stereo matching to compute 3D hairs, and a final cleanup step to remove outliers and refine the recovered 3D hairs. We perform the reconstruction independently for left and right sides of the face and merge the hairs prior to the refinement. To increase robustness we start by reconstructing long hair in a first pass ($>$ 5 mm) and shorter hair in a second.

6.2.1 2D Processing

The first stage of the processing is to detect hair in the images, and seek piecewise linear segments as the first step towards obtaining long fibers of hair. An overview of this process is shown in Figure 6.3.

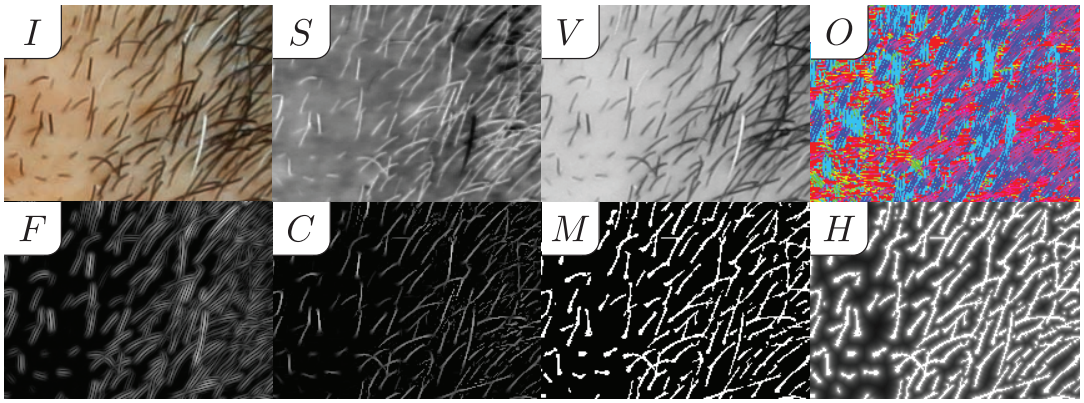


Figure 6.4: Images used during reconstruction - *I: input image, S: saturation channel, V: value channel, O: orientation map, F: Gabor filter response, C: confidence map, M: binary mask, H: hair map.*

(a) Computing a 'Hair Map' for each Image

Our experience is that neither hair color nor hair diameter are uniform across a subject's face. These properties moreover vary along an individual hair, which precludes approaches that assume uniformity. We observed that hair and skin exhibit larger contrast in saturation and value than in hue. Thus images are converted from RGB to HSV space, and the S and V channels are used to discriminate hair. Figure 6.4 shows the original image I in RGB as well as the S and V channels.

Paris et al. [Paris et al., 2004] show that oriented filters are well suited to estimating the local orientation of hair. They employ different filters at multiple scales and determine the best score based on the variance of the filters. This provides an efficient way to estimate a dense orientation field for a dense hair volume. For sparse hair, the situation is different as the hair fibers cover only parts of the image and we need to identify which parts. As we have a good prior on the size of the structural element (the hair thickness) we use only a single oriented filter kernel. An oriented filter kernel K_θ is a kernel that is designed to produce a high response for structures that are oriented along the direction θ when it is convolved with an image. We tried several different filters, such as Gabor and Second Order Gaussians, and found their performance to be very similar in practice. In the following, we use the real part of a Gabor filter. The wavelength λ and standard deviation σ of the filter are set according to the expected hair thickness ($\lambda = 4$, $\sigma = 3$ pixels).

The S and V channels are both convolved with the filter kernel K_θ for different θ (we use 18 different orientations, one every 10 degrees) and the orientation

$\tilde{\theta}$ that produces the highest score $F(x,y) = |K_{\tilde{\theta}} * V|_{(x,y)} + |K_{\tilde{\theta}} * S|_{(x,y)}$ at a pixel (x,y) is stored in the orientation map $O(x,y) = \tilde{\theta}$. As can be seen in Figure 6.4 (F) the filter generates ringing artifacts around the true location of the hair. To suppress these artifacts we propose a non-maximum suppression strategy [Canny, 1986]. The pixel is suppressed unless its score is the maximum score in direction orthogonal to θ over the extent of the filter. The resulting confidence map C is shown in Figure 6.4 (C).

The confidence map C is thresholded using hysteresis [Canny, 1986]. The upper and lower thresholds applied are 0.07 and 0.05. From the binary mask M a hair map H is computed according to $H(x,y) = 1/(1 + d(x,y))$ where $d(x,y)$ is the euclidean distance at (x,y) to the closest foreground pixel in M. The hair map H shown in Figure 6.4 (H) has a value of 1 where hairs are suspected. The value decays quickly when moving away from suspected hair pixels allowing for accurate and robust matching. The hair map is of central importance in matching and growing and is a key difference as opposed to other multi-view stereo systems which usually rely solely on intensity variation. The appearance of hair fibers in different viewpoints can vary substantially due to specularity, translucency, camera focus and of course occlusion through other fibers. The hair map proves to be more reliable under these conditions and permits robust matching of the hair fibers.

(b) Growing Hair in 2D

At the scale we are capturing hair, fibers are essentially one-dimensional structures. Thus the only reasonable neighborhood suited for matching is a one-dimensional neighborhood along the hair itself. We start reconstruction by identifying this neighborhood in an image using a line growing algorithm.

Hair growing in 2D produces a chain of 2D hair segments. A hair segment $s(\mathbf{p}_s, \theta, \ell)$ is a linear segment of length ℓ starting at $\mathbf{p}_s$ in direction of θ. $\mathcal{P}(s)$ denotes the set of pixels covered by the segment s. Growing is based on the hair map H and works as follows. First, a pixel $\mathbf{p}_s = (x_s, y_s)$ in H with $H(\mathbf{p}_s) = 1$ that is not yet part of a hair is selected as seed and the growing direction is determined from the orientation map O. We define a growth cone $\Lambda(r, \gamma)$ with growth resolution r and opening angle 2γ. The cone defines a set of possible next segments that form an angle of less than γ with the axis of the cone and whose distance to the apex is r. See Figure 6.5 for a schematic. The parameters chosen for the growth cone are r = 10 pixels and γ = 60 degrees. When growing a segment, the axis of the growth cone is oriented along the direction θ of the last segment and the apex of the cone is placed at $\mathbf{p}_s$. For

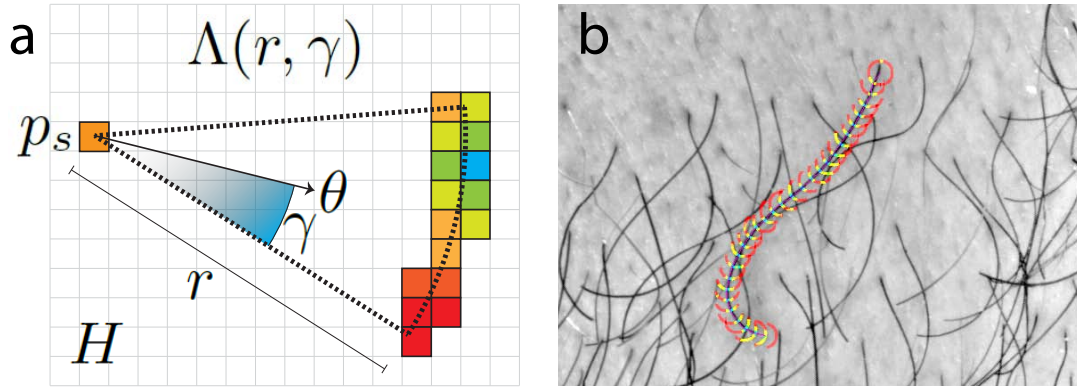


Figure 6.5: 2D hair growth - *Hair growing in 2D makes use of the hair map H. (a) Starting point $\mathbf{p}_s$ and an initial estimate of the growth direction θ are given by the previous segment. The apex of a growth cone $\Lambda(r, \gamma)$ with growth resolution r and opening angle 2γ is placed at $\mathbf{p}_s$ and oriented along θ. For all possible target pixels a score is computed and the pixel with highest score is added to the hair. This process is repeated until the matching score drops below a threshold. (b) An example of a traced hair overlaid the input image.*

each potential growth direction in the growth cone a score is computed as

$$\xi(d\theta) = \left(1 - \frac{|d\theta|}{2\gamma}\right) \psi(\mathcal{P}(s_{d\theta})), \tag{6.1}$$

where $d\theta$ is the angular direction relative to the axis of the cone and $\mathcal{P}(s_{d\theta})$ denotes the set of pixels covered by the segment $s_{d\theta}$. The function ψ is given as

$$\psi(\mathcal{P}) = \frac{1}{\|\mathcal{P}\|} \sum_{\mathbf{p}_i \in \mathcal{P}} \frac{H(\mathbf{p}_i) - \nu}{1 - \nu} \tag{6.2}$$

and will also be used in later sections when operating on the hair map. The parameter ν is defined within $[0, 1[$ and controls how tolerant the score is. The higher ν the more restrictive the score is regarding deviation from the detected hair — but at the same time it will also become less robust. We set $\nu = 0.4$. The pixel $\tilde{\mathbf{p}}$ that produces the highest score is kept as the next segment of the hair and the process is repeated until $\xi(\tilde{\mathbf{p}})$ falls below a given threshold. The threshold was set to 0.5. See Figure 6.5 (b) for an example of a grown 2D hair.

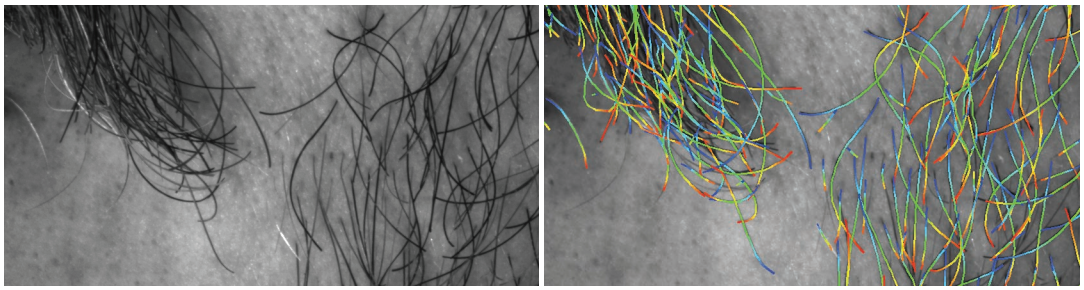


Figure 6.6: *The reconstructed hairs projected into one of the views.*

6.2.2 Matching Hair Segments in 3D

In Section 6.2.1 hair segments were computed separately in each image. This section describes the use of the detected segments in matching across images.

Once the neighborhood has been established, the hair segment is matched in 3D using multi-view stereo. The view in which the 2D hair was grown will be referred to as the reference view $\tilde{c}$. The cameras used for matching are denoted as $\mathcal{C}$. A point $\mathbf{p}$ in the reference view $\tilde{c}$ describes a ray $r(\mathbf{p})$ in space. Given a constrained search space in depth, either computed from the shrink-wrap surface or given by the calibration, the ray $r(\mathbf{p})$ is constrained to a line segment. The projection of this 3D line segment generates an epipolar line segment in every other view $c \in \mathcal{C}$. From the view containing the longest epipolar line segment we find the set of potential 3D positions by creating rays through every pixel on the epipolar line segment and intersecting them with $r(\mathbf{p})$. These potential 3D positions are converted into potential depths d_j along $r(\mathbf{p})$. We then sample the hair segment and compute a matching matrix that contains for every sample $\mathbf{p}_i$ at every depth d_j the highest matching score ζ_{ij}. The matching score is computed from the hair map H using Equation 6.2 and the V-channel of the image as

$$\zeta_{ij} = \left(\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \psi\left(\mathbf{P}_{ij}^{c}\right) \right) \left(1 - \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \left\| V(\mathbf{P}_{ij}^{c}) - V(\mathbf{P}_{ij}^{\tilde{c}}) \right\| \right), \tag{6.3}$$

where $\mathbf{P}_{ij}^{c}$ is shorthand for the projection of the point $\mathbf{P}_{ij}$ into camera c. $\mathbf{P}_{ij}$ is the point on the ray $r(\mathbf{p}_i)$ at depth d_j.

The longest contiguous ridge in the matching matrix is detected and kept as 3D hair consisting of a piecewise linear chain of 3D hair segments. A 3D hair segment is defined either via start and end points as $S(\mathbf{P}_0, \mathbf{P}_1)$ or via length ℓ from P_0 in direction ω as $S(\mathbf{P}_0, \omega, \ell)$. Both notations will be used in the

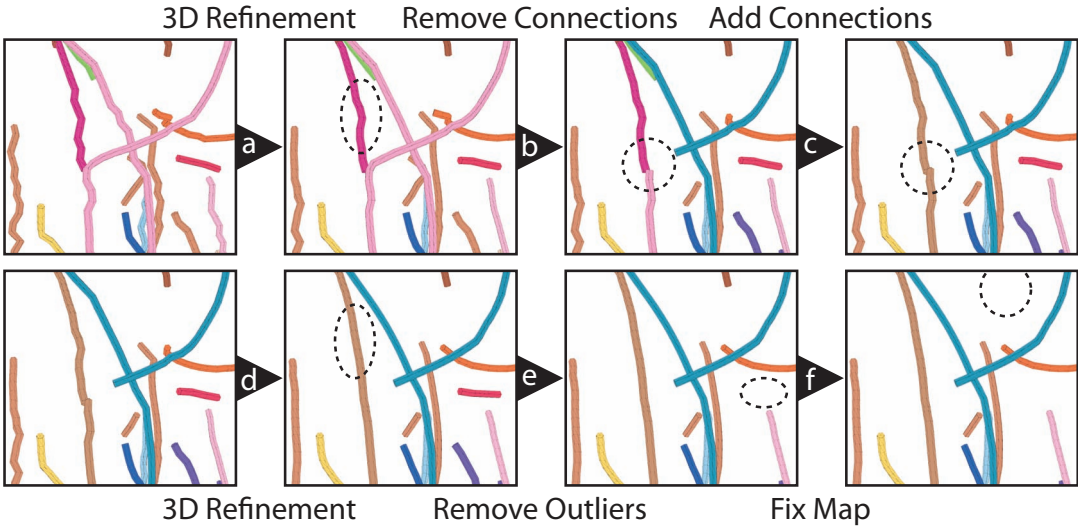


Figure 6.7: *Overview of the refinement and outlier removal steps described in Section 6.2.3.*

following. Figure 6.6 shows the projections of the reconstructed hair into one of the views.

6.2.3 Refinement and Outlier Removal

The processing so far has generated 3D piecewise linear segments for the hairs, with connectivity between the hair segments. Because of the discrete nature of the matching as well as noise and other imperfections, these hair segments are jagged and contain outliers. This section describes a refinement process, including a refinement of the computed 3D data, and a reanalysis of the 3D connectivity. The final step is the removal of 3D outliers. An overview of the individual steps is given in Figure 6.7.

The following steps are carried out in sequence:

(a) 3D Refinement of Computed Hair Segments

This step does a more careful computation of the 3D data for the hair segments, taking the existing 3D hair segments as the start point for the refinement. The computation is a minimization involving a data term that seeks consistency between the images, and a smoothness term that seeks low curvature of the reconstructed hair segment. See Figure 6.8 for a schematic. A point $\mathbf{P}_0$ on the hair with neighbors $\mathbf{P}_-$ and $\mathbf{P}_+$ is refined on the plane normal

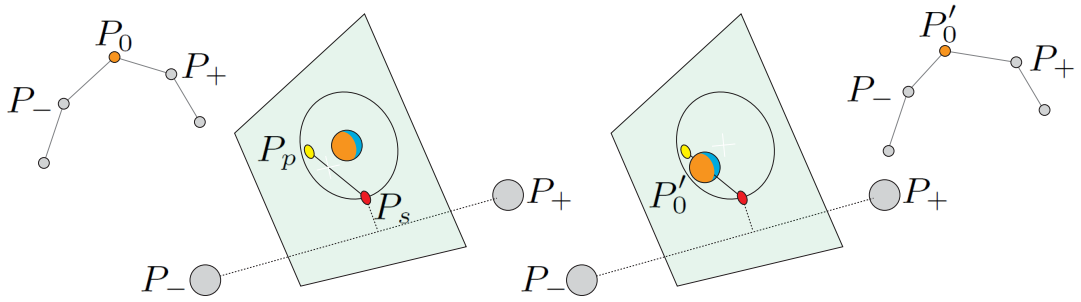


Figure 6.8: The refinement stage - *The point* $\mathbf{P}_0$ *with neighbors* $\mathbf{P}_+$ *and* $\mathbf{P}_-$ *is refined on the plane normal to* $\mathbf{P}_+ - \mathbf{P}_-$*. The refinement computes within a local neighborhood the point* $\mathbf{P}_p$ *that has highest data fidelity and the point* $\mathbf{P}_s$ *that has highest smoothness. The refined position* $\mathbf{P}_0'$ *is computed as the weighted average of these two points as described in Section 6.2.3 (a).*

to $\mathbf{P}_+ - \mathbf{P}_-$. An update is computed as

$$\mathbf{P}_0' = \frac{w\mathbf{P}_p + \lambda\mathbf{P}_s}{w + \lambda}, \tag{6.4}$$

where $\mathbf{P}_p$ denotes the position that highest data fidelity in a local neighborhood defined by the resolution τ and $\mathbf{P}_s$ denotes the position that has highest smoothness given by the projection of $0.5\,(\mathbf{P}_- + \mathbf{P}_+)$ into the neighborhood. λ is a regularization parameter and the weight w is computed as

$$w = \frac{\Xi\left(S\left(\mathbf{P}_-, \mathbf{P}_p\right)\right) + \Xi\left(S\left(\mathbf{P}_p, \mathbf{P}_+\right)\right)}{\Xi\left(S\left(\mathbf{P}_-, \mathbf{P}_s\right)\right) + \Xi\left(S\left(\mathbf{P}_s, \mathbf{P}_+\right)\right)} - 1, \tag{6.5}$$

where Ξ is the matching score defined in Equation 6.6. The refinement is run for 20 iterations with parameters $\tau = 0.1$mm and $\lambda = 0.01$.

(b) Removal of Low-Confidence Connectivity

Due to the projection of the hairs into the image plane, it may happen that the 2D hair tracing algorithm (Section 6.2.1) traces multiple hairs as a single one. If these hairs differ in depth, the 3D matching (Section 6.2.2) will only match and reconstruct one of the hairs. If they do not differ in depth but only in direction, then there will be a point of direction change where one touches the other. This step removes the connectivity of two connected hair segments if the difference in orientation computed by the scalar product is above a fixed threshold (45°).

(c) Addition of New Connectivity

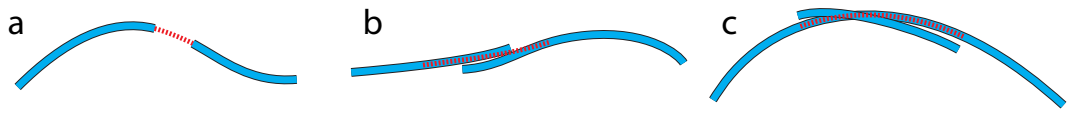


Figure 6.9: Identification of connectivity between hair segments - *(a) Hairs with tips that are spatially close and enclose a small angle are linked. (b) Hairs that have overlapping parts are merged. (c) Hairs that fall completely into an other hair are removed.*

This step involves an explicit search for additional connectivity among the hair segments. Figure 6.9 illustrates the three cases. Firstly, two segments are marked as connected if they satisfy these conditions: the segments have unconnected endpoints with the segments on opposite sides of those endpoints, and the unconnected endpoints are close in space, and the segments have consistent direction. Secondly, two segments are marked as connected if they satisfy the following conditions: the segments have unconnected endpoints with the segments on opposite sides of those endpoints, and the segments are overlapping and have consistent direction. Thirdly, shorter hairs that are completely enclosed by longer hairs are merged into the longer ones.

We allow linking hairs whose tips are closer than 1mm and enclose an angle of $< 20^\circ$. We allow merging segments that are closer than 0.1 mm and enclose an angle of $< 20^\circ$.

(d) Repeat Refinement

Step (a) is repeated.

(e) Removal of 3D Outliers

Outlier removal is done by creating a grid for the 3D workspace, counting the number of hairs in each voxel, and deleting hairs which are distant from the surface and in voxels with a low count (distant from other hairs). This is a basic approach but sufficient for the kind of outliers that are observed—single isolated hairs away from the true surface.

(f) Fix Map

The image structure in the iris or the lips is locally very similar to hair and so the system might reconstruct outlier hairs in these areas. We remove these outliers by drawing two 'Fix Maps', one for a center camera and one for a camera from below. These Fix Maps are very sketchy binary masks that can be created rapidly. Figure 6.10 shows example Fix Maps. Hairs that get pro-

Figure 6.10: Fix Maps - *Fix Maps used to identify areas like eyes or lips where no hairs should be detected. We provide Fix Maps for the two shown viewpoints for all subjects. This is the only manual step in our pipeline and could be automated using feature detectors.*

jected mostly into the masked areas are removed. This is the only manual step in our pipeline and could be automated using feature detectors.

6.2.4 Growing Hair in 3D

Hair growing in 3D is performed in a similar way to hair growing in 2D (Section 6.2.1). A three-dimensional growing cone $\Lambda(r, \gamma)$ is used to determine potential segments. We set the growth resolution to $r = 1\text{mm}$ and $\gamma = 30°$. The apex $\mathbf{P}_s$ of the cone is placed at the tip of the last segment and the axis is aligned with its direction $\omega = (\theta, \phi)$. For every potential segment $S_\omega = S(\mathbf{P}_s, \omega, r)$ of the growth cone with direction $\omega = (d\theta, d\phi)$ a score is computed using all cameras $\mathcal{C}$ where the segment is expected to be visible

$$\Xi(S_\omega) = \frac{\sum_{c \in \mathcal{C}} \rho_\omega^c \psi\left(\mathcal{P}\left(S_\omega^c\right)\right)}{\sum_{c \in \mathcal{C}} \rho_\omega^c} \tag{6.6}$$

where S_ω^c is shorthand for the projection of the segment S_ω into camera c, $\mathcal{P}$ denotes the pixels spanned by this projection and ρ_ω^c denotes the angle the direction ω encloses with the optical axis of the camera c. The growth is terminated when the score value drops below a user defined threshold τ. The effect of varying τ can be seen in Figure 6.11. We used values from 0.1-0.3 for the examples in this chapter.

Figure 6.11: *The effect of varying the threshold τ which terminates the hair growth in 3D. The top row shows the input image (left) and reconstructed hair fibers (right). The center and bottom rows show different growth results obtained by varying the threshold τ.*

6.3 Computing Skin Episurface

The concept of the skin episurface was introduced in Section 6. This section describes its computation, as depicted in Figure 6.12.

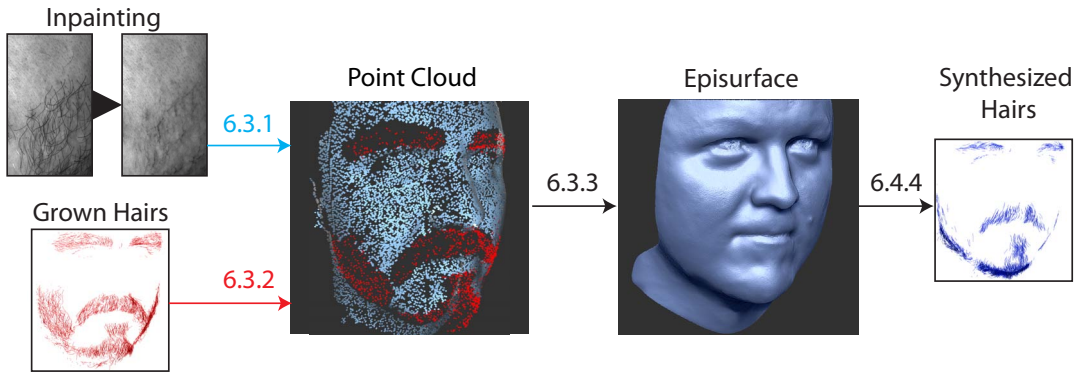


Figure 6.12: Episurface reconstruction overview - *The episurface is reconstructed from the inpainted images (blue) where the surface is visible and from the grown hairs (red) where it is covered with hair. From the episurface, additional hairs can be synthesized.*

6.3.1 Computing Visible Skin Episurface

The part of the skin episurface exposed to the cameras is recomputed using the first part of the method presented in Chapter 5. This part performs pairwise stereo reconstruction and produces a 3D point cloud. The difference from reconstructing the shrink-wrap surface is that we now know where hair is to be expected. Using this knowledge we prepare the data as follows:

- **Masking** - Areas that contain denser hair are masked and therefore excluded from reconstruction. We employ an opening operation followed by a closing operation on the mask to only exclude areas that have considerable amounts of hair. Individual hairs will not be affected as they are removed by the inpainting step.
- **Inpainting** - We employ a Gaussian filter to the image data with spatially varying, anisotropic kernel. The orientation of the filter is given by the orientation map O and the spatial extent σ is computed depending on the hair map H. In areas where H is low, the spatial extent will be low as well ($< 1px$) and no filtering occurs. In areas where H is high the image is blurred, effectively inpainting individual hairs. This

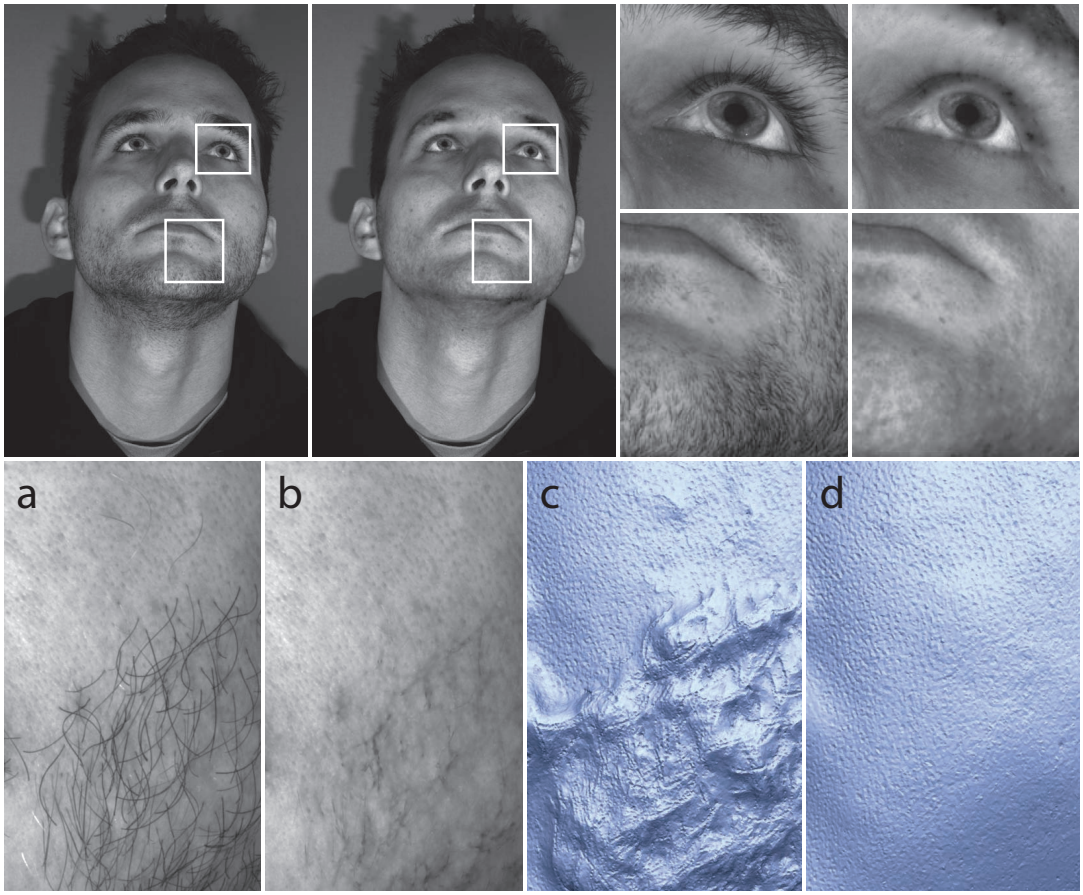


Figure 6.13: Effects of inpainting - *Top row: inpainting reduces the presence of hair while maintaining other facial features. Bottom row: inpainting of sparse hair prevents the stereo reconstruction from creating artifacts in these areas. (a) The captured image; (b) the effect of inpainting as described in Section 6.3.1; (c,d) The reconstructions based on the original and inpainted images. Note how the areas of visible skin are the same, while (d) does not exhibit the shrink-wrap artifacts in areas that contain hair.*

reduces the strong image gradients produced by individual hairs and prevents matching.

The reason for using inpainting instead of masking out individual hairs is that the surface reconstruction from Chapter 5 uses image pyramids to compute the stereo reconstruction. While the images are subsampled linearly, the masks are subsampled using nearest neighbor, which leads to a discrepancy of mask and image at higher layers of the pyramid. Inpainting circumvents this. The effect of inpainting is demonstrated in Figure 6.13.

6.3.2 Estimating Hidden Skin Episurface

The part of the skin episurface that is not exposed to the cameras is estimated in the following way:

- Take each 3D hair computed in Section 6.2. The root of the hair is designated as the end which is furthest below the surface of the shrink-wrap model described in Section 6.1. For short whiskers, both ends of the hair may be close to the surface. In this case, it is arbitrary which end will be chosen as the root, but there is no adverse effect on the subsequent computation. Hairs that are distant from the surface are not considered.
- For each hair root, find neighboring roots within a pre-defined search radius. Do a least-squares fit of a plane to the root and its neighbors, to estimate the surface normal at the root.
- Collect, for all roots, the 3D coordinates and estimated surface normals.

This step produces a point cloud which is a sampling of the underlying hidden episurface.

6.3.3 Computing Skin Episurface

The previous two steps produce two sets of points for the visible and the hidden skin episurface, resp. See Figure 6.14 (b). These are combined and serve as input for the second part of the method from Chapter 5. This part performs a Poisson reconstruction [Kazhdan et al., 2006] followed by a refinement step. We change the refinement step to incorporate the hair map as regularization prior, preferring smoother solutions in areas where hair is expected. Figure 6.14 (d) shows the final episurface compared with the initial shrink-wrap surface.

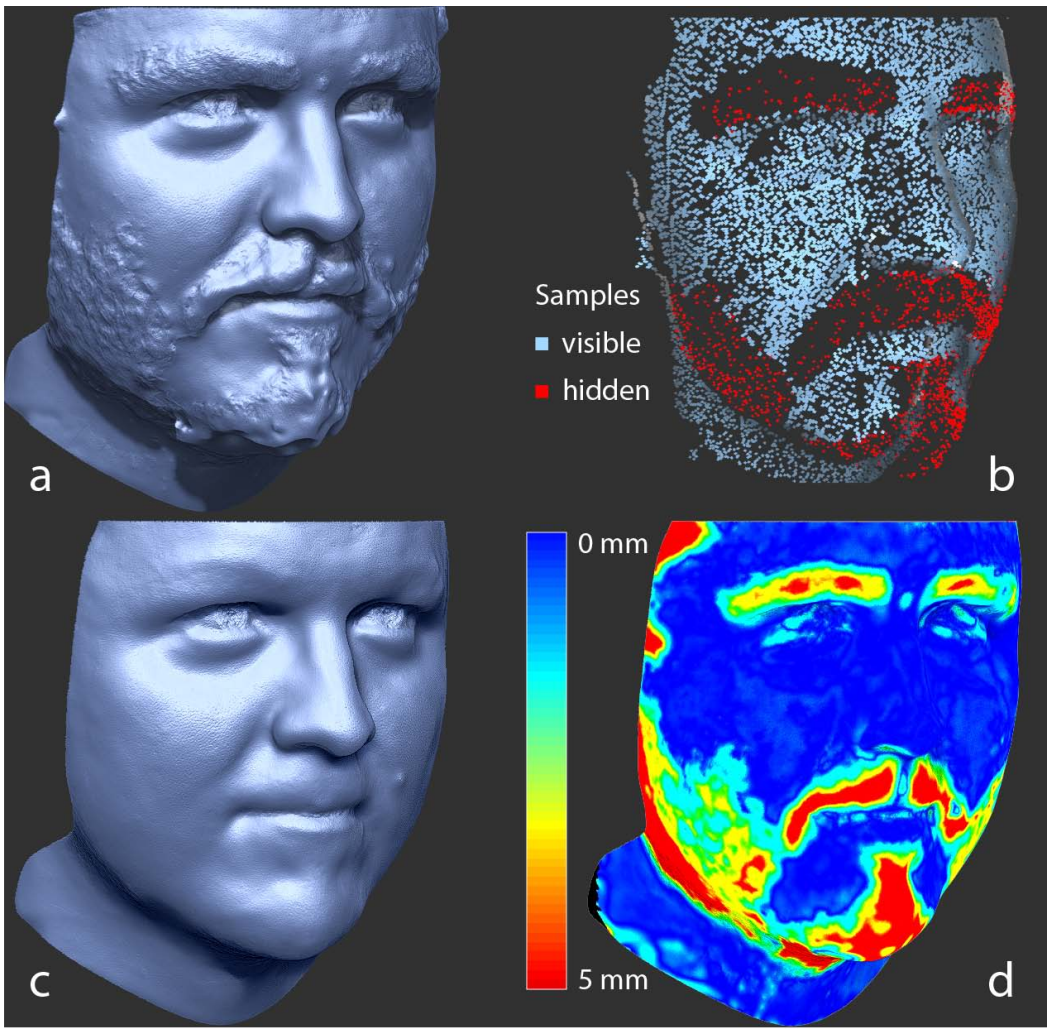


Figure 6.14: The construction of the episurface - *(a) Shrink-Wrap surface produced by the method presented in Chapter 5; (b) the point cloud of the visible episurface in blue (Section 6.3.1) and the points from the hidden episurface in red (Section 6.3.2); (c) the final reconstructed episurface (Section 6.3.3); and (d) visualizes the difference between the two meshes. The two surfaces are almost identical in areas of clear skin but the episurface provides a much smoother hypothesis in areas with hair coverage.*

6.3.4 Synthesizing Hair

This section describes how the 3D hairs computed in Section 6.2 can be augmented in a physically plausible way. The main goal of hair synthesis is to achieve greater density of 3D hair in areas where image data indicates the presence of hair, but individual hairs are indistinct and cannot be reconstructed. Hair synthesis happens in two steps: first, seeds are found on the episurface, and second, the hairs are grown starting from these seeds.

Finding Seed points Finding seed points starts by projecting the reconstructed 3D hairs into the cameras to prevent finding seeds in areas where hair has already been reconstructed. Next every vertex of the episurface is projected into the images. Values $\bar{H}^c$ are computed for all cameras c by averaging the hair map H within a window. The vertex is chosen as hair seed if $\bar{H}^{\tilde{c}} > \alpha$ and $\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \bar{H}^c > \beta$, where $\tilde{c}$ is the camera that has the least foreshortening with the vertex. The parameters α and β are set to 0.3 and 0.2. If a vertex has been selected as seed it will prevent other vertices in its neighborhood from becoming seed points.

Growing Hair The seeds found in the previous step serve as starting points for hair growth. The default growth direction is normal to the surface and the maximal growth length is the average length of all reconstructed sample hairs. These default properties are blended with properties sampled from neighboring hairs that were successfully reconstructed. The properties of the sample hairs and the default growth properties are interpolated using Gaussian RBFs. This leads to a smooth interpolation of growth properties that is faithful to the sample hairs where they could be reconstructed and plausible in areas of dense hair. Using these properties a hair is grown as described in Section 6.2.4 with the difference that the orientation maps are also consulted to give a sense of directionality for the growth. The growth properties are updated as a hair grows to ensure that it remains faithful to the style of the reconstructed fibers within its current neighborhood. The effect of hair synthesis based on reconstructed sample hairs can be seen in Figure 6.15 (c). Hair whiskers in the eyelashes are short and point directly away from the surface, while the hair fibers in the mustache are longer and follow the overall direction in this area.

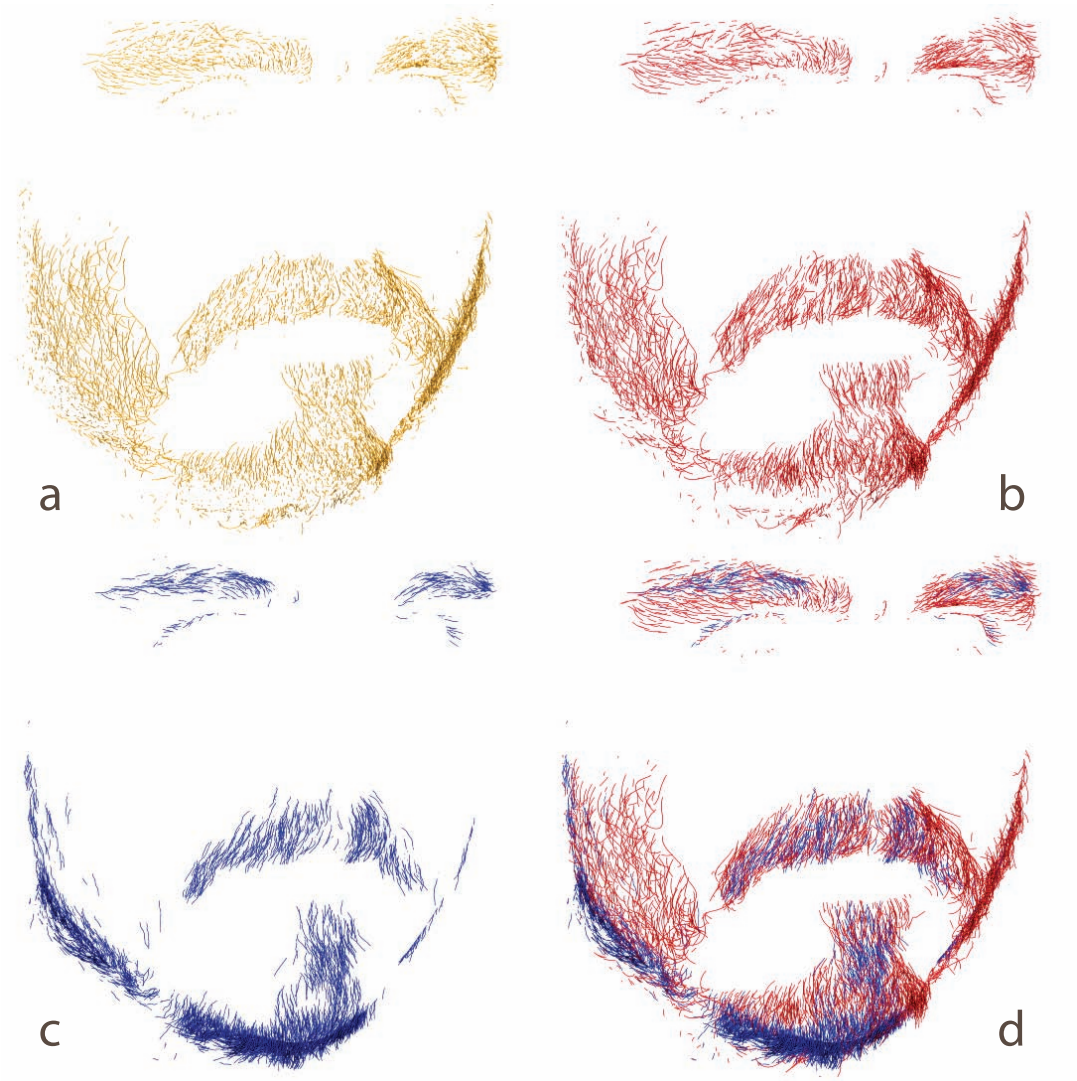


Figure 6.15: The individual stages of the hair reconstruction - *(a) The 3D reconstructed hair after the matching stage (Section 6.2.2); (b) the hair after the growing stage (Section 6.2.4), (c) shows the synthesized hair (Section 6.3.4), and (d) the final composition of (b) and (c).*

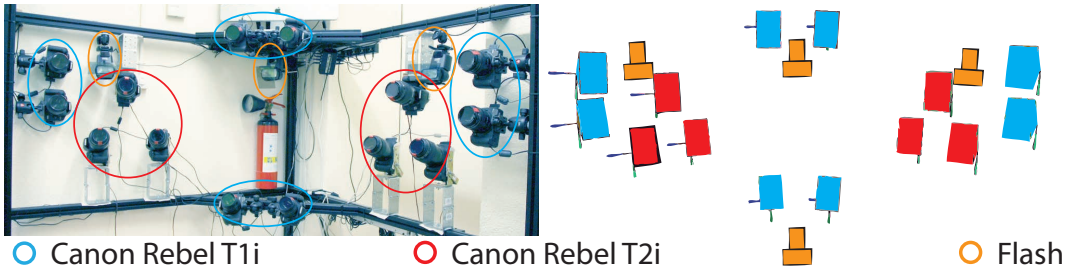


Figure 6.16: Setup used to capture the data - *The setup consists of 8 Canon Rebel T1i cameras equipped with 85mm lenses to capture the full face and 6 Canon Rebel T2i cameras equipped with 100mm lenses to capture the chin area. The subject is illuminated with 4 Canon Flashes (430EX,580EX).*

6.4 Results

Setup We captured all examples using the 14 camera setup presented in Chapter 3 and shown in Figure 6.16. The setup consists of 8 Canon Rebel T1i cameras designated to capture the full face plus 6 Canon Rebel T2i cameras that aim at capturing the chin area with higher resolution. The T1is are equipped with 85mm lenses and are arranged in pairs of two around the subject — one pair on each side, one straight on and one from below. The T2is are equipped with 100mm macro lenses and are arranged in tuples of three on both sides of the subject. The skin reconstruction only uses the T1is while the hair reconstruction makes use of all cameras. The subjects are illuminated with 4 Canon flashes (430EX,580EX) and we use cross polarization to remove specularities. Only a single frame per camera is required and thus the setup can acquire the data within fractions of a second. Reconstruction with our unoptimized prototype pipeline takes approximately 45 minutes on a Mac Pro Desktop computer (8 cores).

Figure 6.17 shows the individual steps of the pipeline for a selected subject. Figures 6.19 and 6.20 show results for various different styles of facial hair demonstrating the robustness of the approach and Figure 6.18 shows selected details and demonstrates the quality of the reconstruction. Figures 6.21 and 6.22 show reconstruction results for a person with both hair stubbles and a goatee beard. Figures 6.23 and 6.24 show reconstruction results for a rather wild mustache. Table 6.1 lists the number of hair fibers reconstructed and synthesized for all reconstructed models.

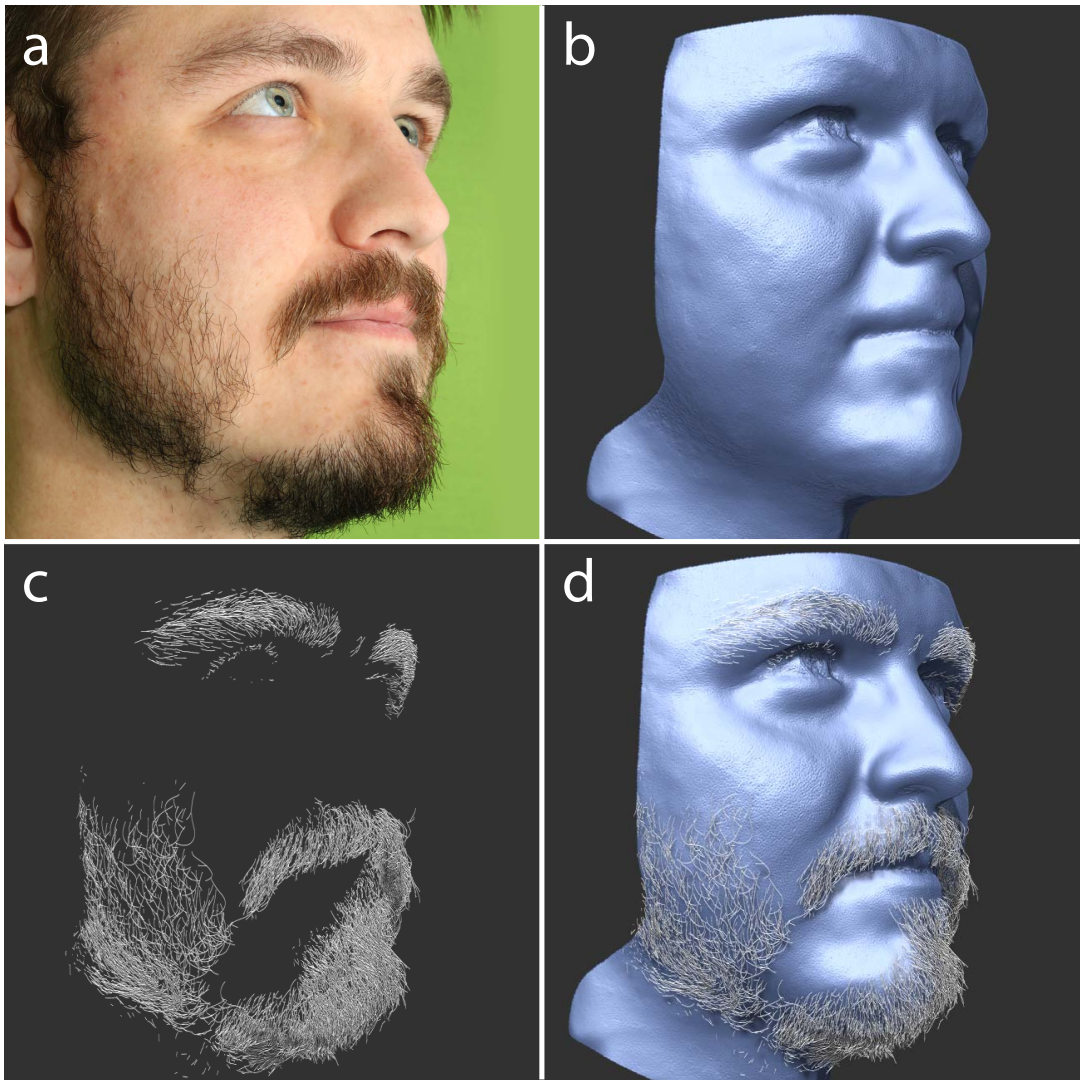


Figure 6.17: Individual steps of the reconstruction pipeline - *(a) Raw image; (b) reconstructed skin episurface; (c) reconstructed hair; (d) skin episurface plus hair.*

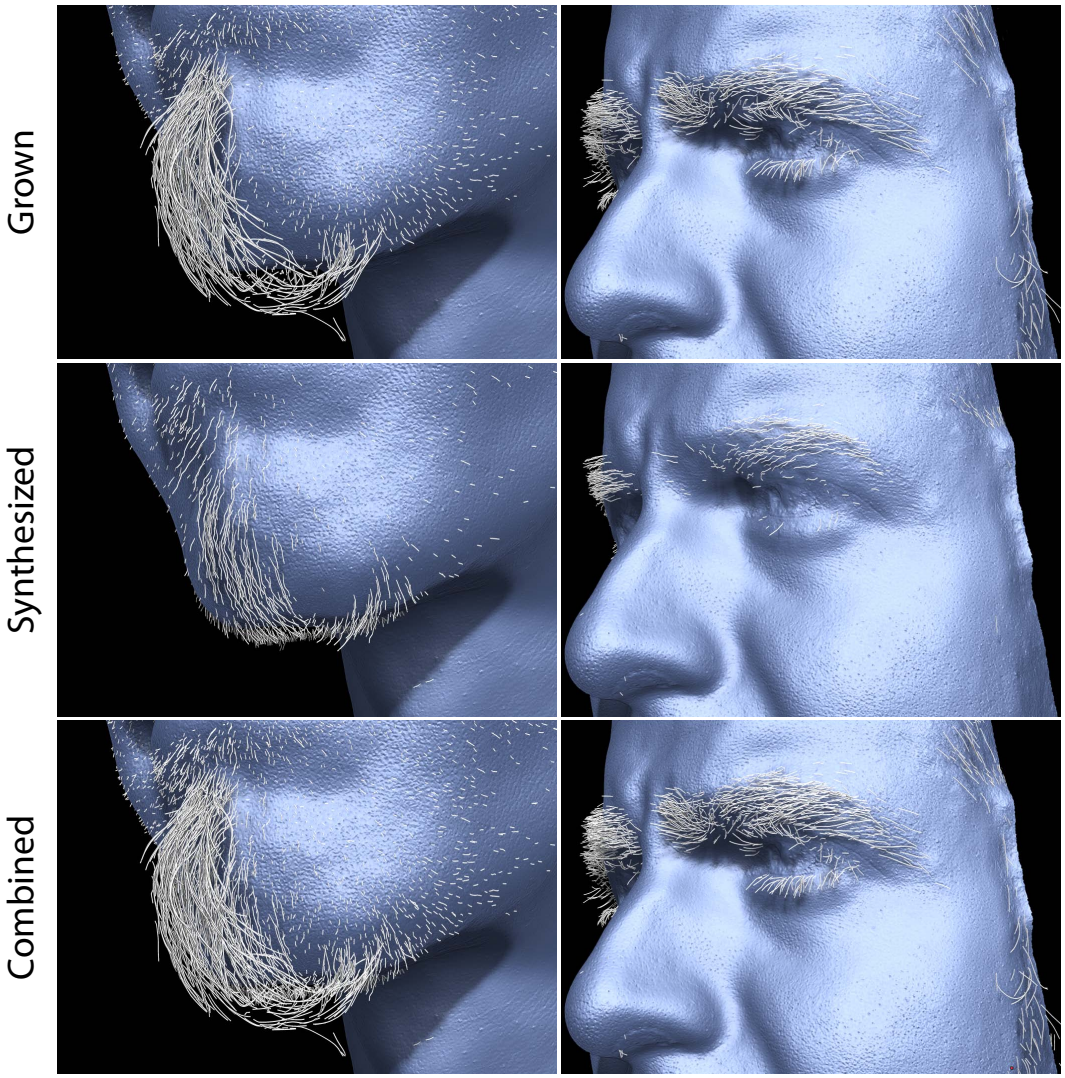


Figure 6.18: *Close-up comparison between different stages of the reconstructed geometry and the real photographs. From left to right: reconstructed hair fibers, synthesized hair fibers, final hair fibers, realistic rendering and real photograph. Note how the overall structure is captured well by the algorithm. Where individual hairs are visible, the algorithm correctly reconstructs them and in areas with dense hair coverage the synthesis provides a plausible volume of hair fibers. Likewise the surface is reconstructed with high quality in areas with no or little hair coverage and provides a plausible substrate in areas with dense hair.*

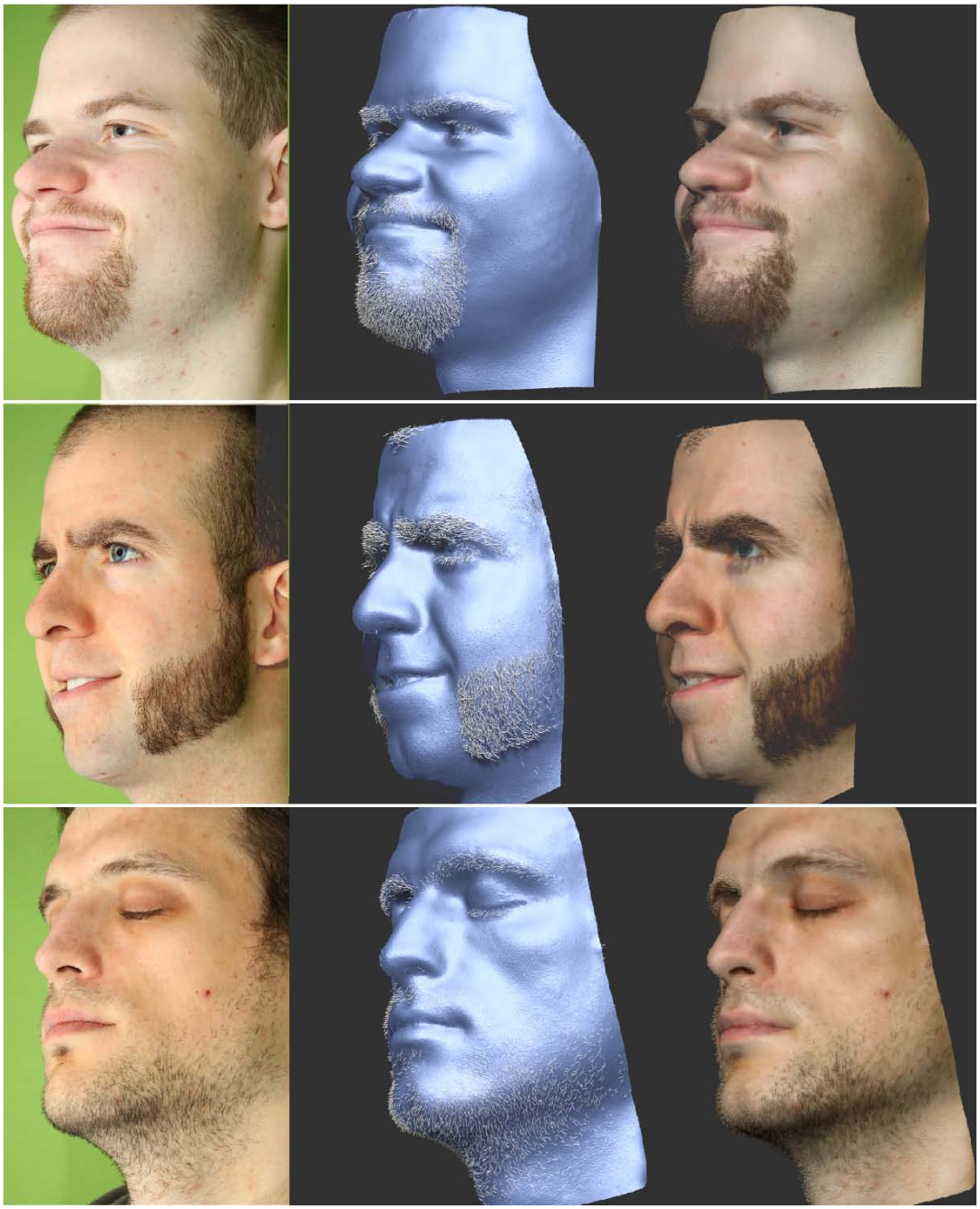


Figure 6.19: *Figures 6.19 and 6.20 show reconstructed models for a variety of subjects demonstrating robust performance for different facial hair stylings.*

Figure 6.20: *Figures 6.19 and 6.20 show reconstructed models for a variety of subjects demonstrating robust performance for different facial hair stylings.*

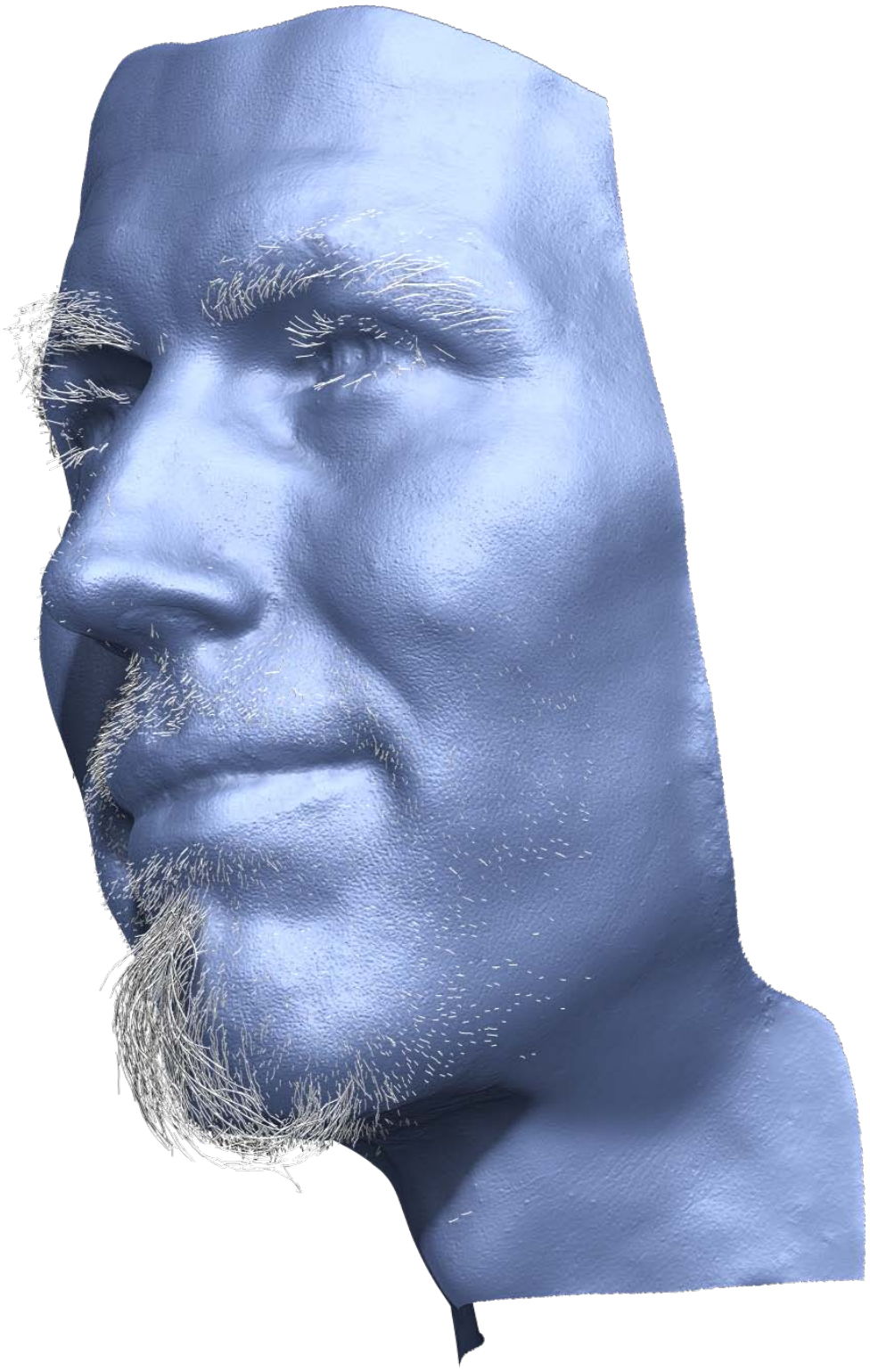
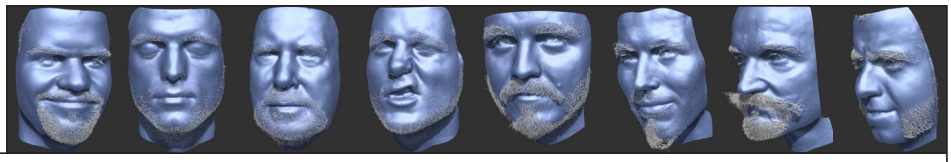


Figure 6.21: *High resolution reconstruction of a goatee beard.*

Figure 6.22: *The same scan as shown in Figure 6.21 but with applied texture. Texture is recovered from the same images that are used for reconstruction and is thus inherently compatible with the scan.*

Figure 6.23: *High resolution reconstruction of a mustache.*

Figure 6.24: *The same scan as shown in Figure 6.23 but with applied texture. Texture is recovered from the same images that are used for reconstruction and is thus inherently compatible with the scan.*



Recon.	4160	5003	4193	5811	4565	3552	6851	2779
Synth.	104	179	694	973	4963	1831	494	654

Table 6.1: *Number of hair fibers reconstructed (top row) and synthesized (bottom row) for all reconstructed models. The amount of synthesized hair is typically low, except for subjects that have dense hair coverage.*

Finally Figure 6.25 shows the reconstructed episurface for a shaped tufty beard. The shape approximates the beard rather than the underlying skin surface in this case. This is not an unexpected result because the episurface is not formulated with the idea of recovering underlying face shape. It would be an interesting line of future research to provide episurfaces that have the goal to be anatomically plausible.

6.5 Conclusion

We have presented an algorithm for face capture that successfully reconstructs facial hair fibers as well as the face's underlying skin surface. We show that treating skin and hair in a coupled fashion delivers accurate reconstruction in areas of high visibility, and gives plausible results in areas of dense occlusion. We demonstrate reconstructions of a number of individuals exhibiting a variety of facial-hair styles. The impact of our work is reflected by the significance of facial hair in our cultural heritage. Our image of many historic figures is dominated by their facial-hair features, including Albert Einstein's bushy white whiskers, Abraham Lincoln's characteristic beard, and Salvador Dali's distinctive mustache [Dalí and Halsman, 1954] (now eponymously known as a "dali"). Today, facial hair remains at the core of individual expression, as evidenced by the ever changing popularity of different facial-hair styles. Our work provides a means to capture this piece of popular culture for use in contemporary applications as well as accurate preservation for future generations.

Limitations of our system direct us to areas of future work. Our algorithm delivers the best results in the presence of short, sparse hairs. In areas of dense hair, such as a long, thick beard, the degree of occlusion can be

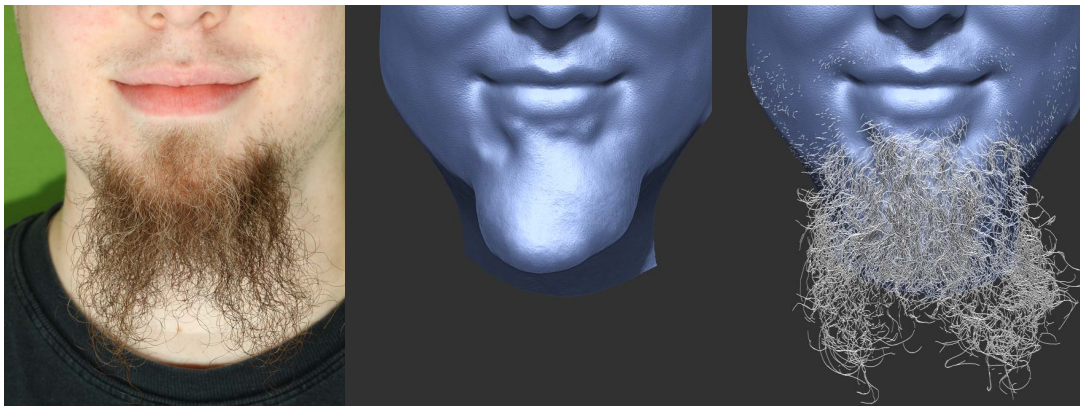


Figure 6.25: *Reconstruction of the skin episurface for a tufty beard. Note that the skin episurface is a pseudo-surface underneath the uppermost layer of hair and is not expected to approximate the shape of the underlying skin in the case of protruding facial hair.*

so great that accurate hair and skin reconstruction is not feasible. Future work could combine algorithms that target long, whole-head hair styles [Paris et al., 2004, Wei et al., 2005b, Paris et al., 2008] with our facial hair-system to deliver high-quality results even for long, thick beards, mustaches, or side burns. Preliminary tests on synthetic hair (Figure 6.26) show that the presented method works well with long hair fibers, which would allow reconstructing the outermost hair layer of whole-head hair styles. There could be a tremendous benefit in terms of overall realism in reconstructing the outer hair layer exactly using our method, and augmenting this result with the powerful estimation methods of whole-head hair capture systems. Our hair-detection algorithm is limited by the amount of contrast in the camera images, and skin-colored hairs may be missed. Likewise, dynamic range can be an issue in cases where the skin and hair vary greatly in brightness (a white beard on black skin, or vice versa) due to exposure limitations. Extending our current algorithm to be robust against contrast and dynamic range limitations is another area of future work. We focus on geometry capture, and only include a limited amount of color information in our results. A thorough treatment of skin and hair appearance capture under varying lighting conditions offers many challenges for future work.

Other hair features such as shape or thickness could also be reconstructed and incorporated in our example-based hair-synthesis algorithm.

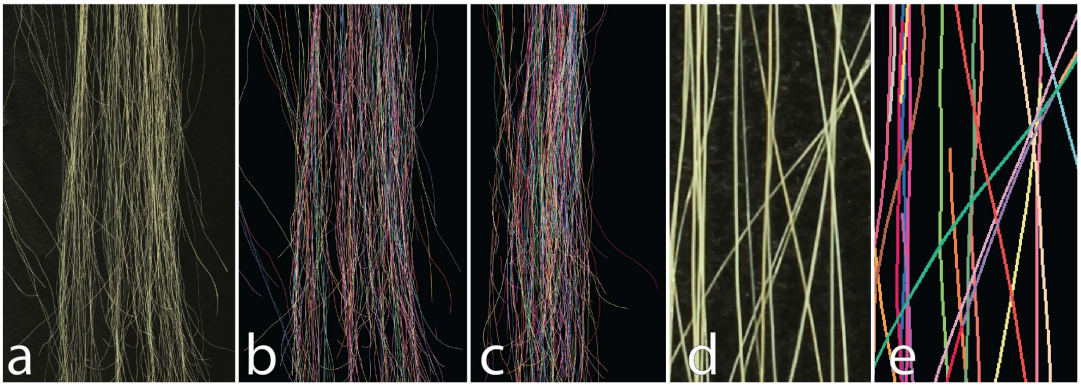


Figure 6.26: Synthetic hair test - *Given a sample of long synthetic hair (a), the system reconstructs the 3D hairs shown in (b) and (c). A close-up of the input image and the corresponding reconstructed hairs are shown in (d) and (e). Notice how most fibers are correctly reconstructed.*

Extending the work to capture hair dynamics would be a challenging and very interesting line of future research. Spatio-temporal capture and reconstruction would on the one hand increase the complexity of the problem, but on the other hand also introduce additional data and constraints that could be leveraged. A further topic of future research is given by the episurface concept as it would be interesting to investigate ways to provide episurfaces which are anatomically correct. This could be achieved by incorporating prior knowledge, e.g. in the form of a morphable model [Blanz and Vetter, 1999].

Perhaps the most exciting area of future work lies in extending hair and skin surface capture beyond the face, to include the entire human body. Doing so will permit the capture of the human form at a level of fidelity not yet possible. An even more far-reaching goal lies in moving beyond humans to other species. A characteristic feature of all mammals is the presence of hair. This huge range — from the soft coat of a cat, to the wiry bristles of an elephant, to the dense, waterproof fur of a sea otter — provides an exciting and compelling long-term target for future work in hair and skin surface capture.

Part III

Motion

C H A P T E R 7

Performance Capture

The algorithm presented in Chapters 5 and 6 permit to reconstruct the 3D geometry of human faces at very high spatial resolutions. The algorithms are single shot — thus requiring only a single exposure per reconstruction. However, they are static snapshots and do not contain any temporal information whatsoever. As motivated in Chapter 1, most areas interested in synthetic human faces, such as computer-generated animation, special effects, games, interactive environments, synthetic storytelling, and virtual reality, require not only high resolution 3D geometry but also accurate facial motion. A performance capture result must thus exhibit both a great deal of spatial fidelity and temporal accuracy in order to be an authentic reproduction of a real actor's performance. Numerous technical challenges such as robust tracking of facial features under extreme deformations and error accumulation over long capture sessions contribute to the problem's difficulty.

Building on the static high resolution 3D skin reconstruction system presented in Chapter 5, this chapter introduces a reconstruction algorithm that delivers a single, consistent mesh deforming over time to precisely match an actor's performance. Our results demonstrate that our system is robust to expressive and fast facial motions, reproducing extreme deformations with minimal drift. Our system requires no makeup so that temporally varying texture can be derived directly from the captured video. And, the computa-

tion is parallelizable so that long sequences can be reconstructed efficiently using a multi-core implementation.

Our high-quality results derive from two technical innovations. First, we employ a robust tracking algorithm that integrates tracking in image space and uses the integrated result to propagate a single reference mesh to each target frame. This strategy yields results superior to mesh-based tracking techniques for a number of reasons: (a) The image data typically contains much more detail, facilitating more accurate tracking. (b) The problem of error propagation due to inaccurate tracking in image space is dealt with in the same domain in which it occurs. (c) There is no complication of distortion due to parameterization, a technique used frequently in mesh processing algorithms. Additionally, because the image-space tracking is computed for each camera, multiple hypotheses are propagated forward in time. If one flow computation develops inaccuracies, the others can compensate.

Although our image-space tracker is accurate for short sequences, the eventual accumulation of integration error when reconstructing long capture sessions is unavoidable unless special care is taken. Our second contribution addresses this issue by employing an "anchor frame" concept that relies on the observation that a lengthy facial performance will contain many frames that are similar in appearance. For example, when speaking, the face naturally returns to the resting pose between sentences or during speech pauses. Our method defines one frame as a reference frame and then marks all other frames similar to the reference as anchor frames. Due to the similarity, our image tracker can compute the flow from the reference to each anchor independently and with high accuracy. Our system can then treat each sequence between two consecutive anchors independently, integrating the tracking from both sides and enforcing continuity at the anchor boundaries. The accurate tracking of each anchor frame prevents error accumulation in lengthy performances. And, since the computation of the track between two anchors is independent, the algorithm can be parallelized across multiple cores or CPUs. Our method can use anchor frames that span multiple capture sessions of the same subject on different occasions without any additional special processing. This can be used to "splice" and "mix and match" unrelated clips, adding a powerful new capability to the editorial process.

7.1 Reconstruction Pipeline

This section describes our method for passive facial performance capture. The input is a sequence of frames of the face, where a 'frame' is a set of n

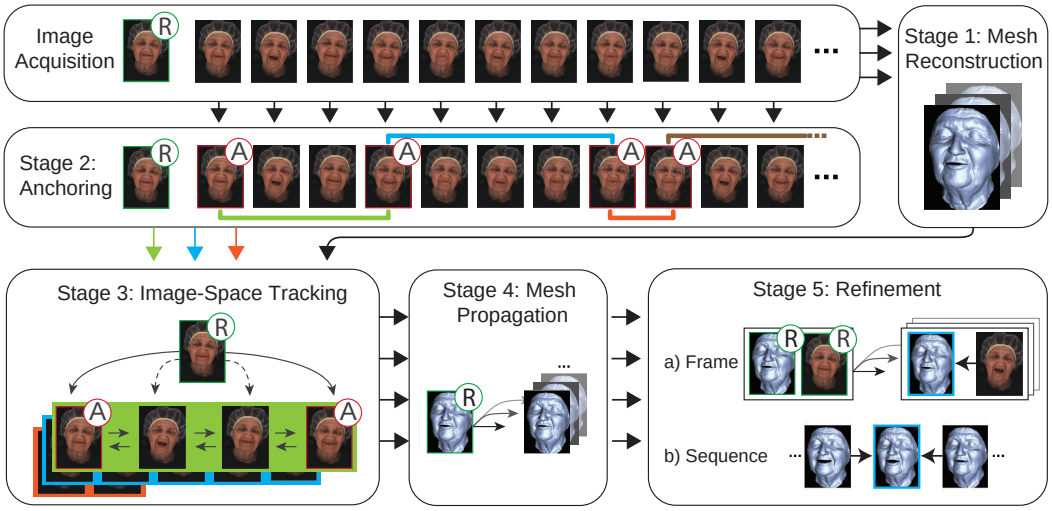


Figure 7.1: System overview - *Image acquisition is followed by mesh reconstruction (Stage 1) and anchor frames are detected to partition the sequence (Stage 2). The image-space tracking step matches the reference frame to all frames in the sequence (Stage 3), and then the reference mesh is propagated to each frame (Stage 4). Finally, the meshes are refined for a high-quality result (Stage 5).*

images acquired at one timestep. The output is a sequence of 3D meshes, one per frame, which move and deform in correspondence with the physical activity of the face. Each mesh vertex corresponds to a fixed physical point on the face and maintains that correspondence throughout the sequence within the bounds of error. Computation of the output meshes takes into account the image data, a prior on spatial coherence of the surface, and a prior on temporal coherence of the dynamically deforming surface. Figure 7.1 illustrates the five stages in the method:

- **Stage 1: Computation of Initial Meshes** - Each frame in the sequence is processed independently to generate a first estimate of the mesh for that frame.
- **Stage 2: Anchoring** - One frame is manually identified as the reference frame (marked "R" in Fig. 7.1). Frames with similar image appearance (similar face expression and head orientation) are detected automatically and labelled as anchor frames (marked "A" in Fig. 7.1 and Fig. 7.2). Anchor frames will provide a way to partition the complete sequence into clips of frames for the processing in Stage 3.

- **Stage 3: Image-Space Tracking** - The goal of this stage is to track image pixels from the reference frame to each frame in the sequence. The process starts by tracking image pixels from the reference frame to the anchor frames, which is straightforward because the image appearance in both frames is, by definition, similar. This will guide the tracking of image pixels from the reference frame to all other frames in the sequence. Tracking to the non-anchor frames is performed sequentially, starting from the nearest anchor frames.
- **Stage 4: Mesh Propagation** - The tracked image pixels obtained in Stage 3 provide a way to propagate the reference mesh, which is the mesh computed for the reference frame in Stage 1, to all frames in the sequence. We use the term *mesh propagation* of the reference mesh to mean the computation of new positions in space of the mesh vertices, in correspondence with physical movement of the face.
- **Stage 5: Mesh Refinement** - Previous stages generated a propagation of the reference mesh to each frame in the sequence. This deforming mesh sequence provides an initial estimate of the face motion, which is refined to enforce consistency with the image data while applying priors on spatial and temporal coherence of the deforming surface.

The individual stages are described in more detail below.

Terminology - A frame $\mathcal{F}^t$ is the collection of the images I_c^t of all cameras $c \in \mathcal{C}$ at time t. Tracking image pixels from frame $\mathcal{F}^t$ to frame $\mathcal{F}^{t'}$ will be shorthand for tracking image pixels in each image pair $(I^t, I^{t'})_c$ of all cameras $c \in \mathcal{C}$.

7.1.1 Stage 1: Computation of Initial Meshes

We begin by recording an actor's performance from n different viewpoints, captured with uniform illumination, as described in Chapter 3. Each frame in the sequence is processed independently to generate a mesh for the face using the 3D skin surface reconstruction method proposed in Chapter 5. This gives us single-shot geometry for each frame with visually realistic pore-level details. There is no temporal correspondence in the resulting sequence of meshes, i.e. their mesh structure (number of vertices and triangulation structure) is totally unrelated. The goal of subsequent stages will be to generate a mesh sequence that is compatible, i.e. share a common vertex set and mesh structure, based on the initial meshes.

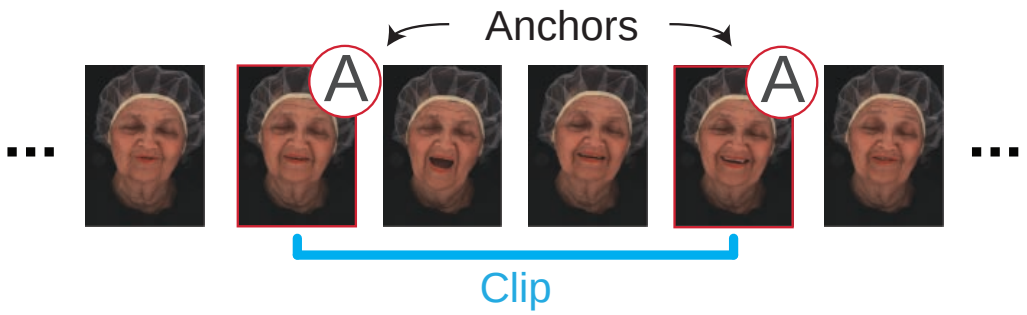


Figure 7.2: Anchor frames - *Frames that are similar to the reference frame are labelled as anchors. A clip is a sequence of frames bounded by two anchor frames (an anchor frame might be both the last frame of one clip and the first frame of the next clip).*

7.1.2 Stage 2: Anchoring

One frame in the sequence is identified as the reference frame. This is currently performed manually, however this step could be automated by analyzing the data and then automatically picking a repetitive pose. Frames with similar image appearance (similar face expression and orientation) are detected and labelled as anchor frames. Anchor frames are used to partition a sequence of frames into clips as shown in Figure 7.2. For example, the reference frame can be chosen with the face in its natural rest expression, and this will produce anchor frames along the sequence whenever the face returns to a rest (or similar) expression. Anchoring will be utilized in Stage 3 where the image-space tracking operates at the level of clips, eliminating the need to track a lengthy sequence of frames.

7.1.2.1 Motivation

Drift (or error accumulation) in tracking is a key issue when processing a long sequence. Anchor frames provide a way to decompose the sequence into clips, effectively allowing multiple starting points for the processing but still having a common root in the reference frame. An immediate benefit of this technique is that it naturally allows parallelization of the computation. But the more important benefit is that it results in periodic "resets" that prevent the accumulation of drift when tracking along the sequence. The technique also confines catastrophic failures in tracking to the frames in which they occur, which could arise from occlusion, motion blur, or the face outside the image.

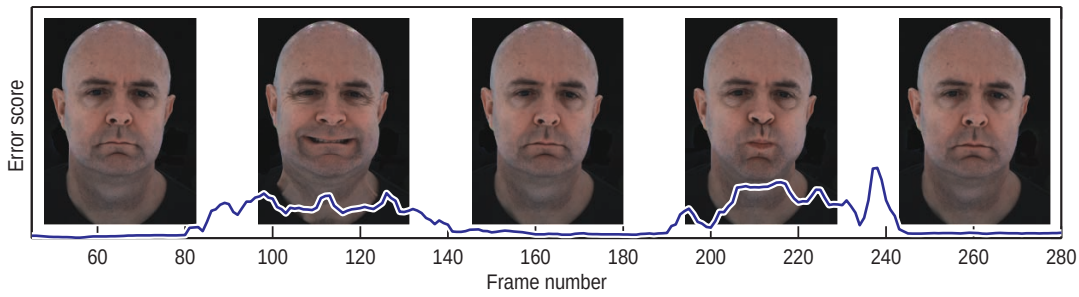


Figure 7.3: Anchoring - *The reference frame is compared with all frames in the sequence. Frames where the image appearance is similar are labelled as anchor frames.*

7.1.2.2 Identifying Anchor Frames

Our method for using the reference frame to determine anchor frames proceeds as follows:

- Detect a feature set $\mathcal{S}_c$ in image I_c^R in the reference frame, for all cameras $c \in \mathcal{C}$. There is no requirement to detect the same physical face features in the different images. Our features correspond to a uniform sampling of the images with a sample rate of 0.05 (every 20th pixel). While we found these simple features sufficient, more sophisticated features such as SIFT [Lowe, 2004] could be used.
- Perform correspondence matching of S_c between I_c^R and I_c^t in the target frame $\mathcal{F}^t$, for cameras $c \in \mathcal{C}$. We employ normalized cross-correlation with 9×9 windows centered on each feature pixel.
- Compute an error score E as the sum of the cross-correlation scores over all features in all feature sets $\mathcal{S}_c$. Label the target frame as an anchor frame if E is below a predefined threshold.

Figure 7.3 shows the variation in E for a few example frames.

7.1.3 Stage 3: Image-Space Tracking

The goal of this stage is to track image pixels from the reference frame to each frame in the sequence. As stated previously, tracking pixels between frames is shorthand for tracking pixels in each camera in the frames. The basic method of finding correspondence is block-based matching using normalized cross-correlation.

However, there are two distinct situations for matching. The anchor frames identified in Stage 2 have, by definition, similar image appearance to the reference frame. Thus image correspondence is straightforward. The unanchored frames, however, may contain quite different facial expressions from the reference face and cannot be reliably matched directly.

Matching is done independently per clip, using the clips that were generated in Stage 7.1.2. Correspondence from the reference frame is first obtained for the anchor frames, as described in Section 7.1.3.1. These correspondences are then propagated from the anchor frames bounding the clip to the intermediate unanchored frames within the clip, both forwards and backwards, as described in Section 7.1.3.3.

7.1.3.1 Tracking from Reference Frame to Anchor Frames

The orientation and expression of the face in anchor frames is similar to that in the reference frame, but its location within the frame might differ substantially. Thus the matching uses an image pyramid to detect large motions. The process starts at the coarsest level of the pyramid and matches a sparse set of features (we again use uniform image samples) to estimate extremal motions $(m_x, m_y)^{\pm}$, followed by the dense motion estimation method described in Section 7.1.3.2 using a search window of size $[(m_x^+ - m_x^-) \times (m_y^+ - m_y^-)]$. The resulting motion field is upsampled to the next higher resolution and the dense motion estimation is repeated but now with a search window of fixed size (3×3). This is repeated until the highest resolution layer is reached. This is done for each anchor frame to provide the motion fields $\mathbf{u}_c^{R \to A}$ from the reference frame R to anchor frame A for cameras $c \in \mathcal{C}$.

7.1.3.2 Dense Motion Estimation

The motion estimation is an extension to 2D of the matching described in Chapter 5 and has the following steps:

Matching A pixel $\mathbf{p}$ in image I_c^R is matched to its best match $\mathbf{q}$ in image I_c^A using 3×3 block-based normalized cross-correlation with the search window introduced in 7.1.3.1. This provides the forward motion estimation $\mathbf{u} = \mathbf{q} - \mathbf{p}$. The matching is run also in the reverse direction from I_c^A to I_c^R starting from $\mathbf{q}$ to provide the backward motion estimation $\mathbf{v} = \bar{\mathbf{p}} - \mathbf{q}$, where $\bar{\mathbf{p}}$ is the pixel in I_c^R that backward matches $\mathbf{q}$.

Filtering A match is not accepted for pixels where $\|\mathbf{u} + \mathbf{v}\|$ is larger than a threshold (one pixel).

Re-Matching Unmatched pixels are re-matched using accepted neighbor matches for guidance. This process is iterated until all unmatched pixels are matched.

Refining The computed matches are refined by combining two terms, one for photometric consistency of the match in the two images, and one which uses a depth map to prevent smoothing over depth discontinuities. Depth maps are obtained from the meshes computed in Stage 1, reprojected back onto the images.

The formulation to refine the motion is a convex combination $\mathbf{u}' = (w_p\mathbf{u}_p + w_s\mathbf{u}_s)/(w_p + w_s)$. The weights w_p and w_s and the photometric term $\mathbf{u}_p$ are the same as in Chapter 5, but the regularization term $\mathbf{u}_s$ is modified based on the depth map δ and the matching error ξ

$$\mathbf{u}_s(\mathbf{p}) = \sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} w_{\mathbf{p}'}\mathbf{u}_{\mathbf{p}'}, \tag{7.1}$$

where $w_{\mathbf{p}'} = \exp\left(\frac{\|\delta_{\mathbf{p}'} - \delta_{\mathbf{p}}\|^2}{\sigma^2}\right)(1 - \xi_{\mathbf{p}'})$ and $\mathcal{N}(\mathbf{p})$ denotes the neighborhood of $\mathbf{p}$, i.e. the 8 neighboring pixels. The value of σ is 1mm in our experiments.

7.1.3.3 Tracking from Reference Frame to Unanchored Frames

Direct tracking of pixels from the reference frame to the unanchored frames is not reliable because image appearance can differ substantially between the two. Instead the matching between the reference frame and the anchor frames obtained in Section 7.1.3.1 is used to aid the process. Frames are tracked incrementally within a clip starting from the relevant anchor frames. The pixel tracking information from the reference frame to the anchor frame, plus the incremental frame-to-frame matching, is used to infer the pixel tracking from the reference frame to the individual unanchored frames.

$$\mathbf{u}_c^{R\to t} = \mathbf{u}_c^{R\to A} + \sum_{i<t} \mathbf{u}_c^{i\to i+1}. \tag{7.2}$$

This generates a motion field for each unanchored frame, which is refined as described in Section 7.1.3.2. This last step allows the motion field to self-correct for small drift with respect to the reference frame (track-to-first principle).

Each clip is bound by a start and end anchor frame, and the pixel tracking above is done from the start anchor in the forward direction and from the end anchor in the backward direction. The forward and backward motion fields may differ due to error. This is resolved by computing an error field for each motion field, smoothing the error fields to remove local perturbation, and then taking the lowest smoothed error to obtain the best match at each pixel. Individual pixels may vary in their assignment of either forward or backward propagation, however the local error (and thus, assignment of propagation) tends to be temporally coherent. Any inconsistencies are resolved in the refinement process described in Stage 5.

7.1.4 Stage 4: Mesh Propagation

The reference mesh consists of a set of vertices $\mathbf{x}_i^R$ for the reference frame, obtained in Stage 1. Each vertex represents a physical point on the face. The goal of mesh propagation to a frame $\mathcal{F}^t$ in the sequence is to find the transformed 3D position $\mathbf{x}_i^t$ of each vertex $\mathbf{x}_i^R$ due to the motion and deformation of the face from the reference frame to frame $\mathcal{F}^t$.

Stage 3 has provided the motion fields from reference frame $\mathcal{F}^R$ to $\mathcal{F}^t$. The method for using the motion fields to estimate the propagated vertices $\mathbf{x}_j^t$ is similar to [Bradley et al., 2010]. Each vertex $\mathbf{x}_i^R$ is projected onto the camera images in $\mathcal{F}^R$, and the corresponding motion vectors from the per-camera motion fields are applied. Back-projecting from the new pixel locations onto the initial geometry for frame $\mathcal{F}^t$ (obtained in Stage 1) gives a per-camera estimate of the propagated 3D position. Estimates are weighted by the dot product between the surface normal and the camera view vector. Spatial clustering is used to identify outliers, and then the final estimate is obtained from a weighted average within the best cluster.

Mesh propagation is now complete and vertices $\mathbf{x}_i^R$ are in correspondence with vertices $\mathbf{x}_i^t$ in every frame $\mathcal{F}^t$.

7.1.5 Stage 5: Mesh Refinement

Previous stages have generated a propagation of the reference mesh to each frame in the sequence. This is a step closer to the goal stated earlier, to have temporal correspondence of meshes along the sequence. However the propagated meshes have been computed with different methods (for anchor frames and unanchored frames) and computed independently for each timestep. The refinement described in this section updates the meshes to ensure a uniform treatment in the computation of all frames and to apply temporal coherence. There are two stages—a refinement that acts independently on each frame and can thus be parallelized, and a refinement that aims for temporal coherence between frames.

7.1.5.1 Refinement per Frame

The goal of the per-frame refinement is to find for each vertex the position in space that optimizes the following objectives:

- **Spatial image fidelity** - The reprojections in all visible cameras for frame $\mathcal{F}^t$ should be similar.
- **Temporal image fidelity** - The reprojections in frames $\mathcal{F}^t$ and $\mathcal{F}^R$ for each visible camera should be similar.
- **Mesh fidelity** - The transformed mesh $\mathcal{M}^t$ should locally be similar to the reference mesh $\mathcal{M}^R$.
- **Geometry smoothness** - The transformed geometry should be locally smooth.

To render the process robust and efficient we follow the proposition of Furukawa *et al.* [Furukawa and Ponce, 2009b] and treat motion and shape separately. The refinement is an iterative process in 2.5D that interleaves motion and shape refinement.

Shape Refinement Shape is refined along the normal. We employ the shape refinement framework from Chapter 5, which aims to find vertex displacements $\mathbf{x}'$ that jointly satisfy photometric consistency constraints, surface smoothness constraints, and mesoscopic position estimates,

$$\mathbf{x}' = (w_p\mathbf{x}_p + w_s\mathbf{x}_s + w_\mu\mathbf{x}_\mu)/(w_p + w_s + w_\mu). \tag{7.3}$$

We use the variation proposed in section 5.2.4 to produce smoother solutions in areas of higher matching error (eye-brows, nostrils, etc.). For all examples in this chapter we use $\lambda^s = (0.03, 0.7, 1000)$.

Motion Refinement Motion is refined in the tangent plane of each vertex. Similar to the shape refinement process, we find vertex displacements $\mathbf{x}'$ that jointly satisfy photometric consistency and mesh regularization,

$$\mathbf{x}' = (w_p \mathbf{x}_p + w_s \mathbf{x}_s)/(w_p + w_s). \tag{7.4}$$

In this case, the photometric position estimate $\mathbf{x}_p$ is the position on the tangent plane that maximizes photometric consistency between current and reference frame. We use normalized cross-correlation as a measure of consistency and compute it by reprojecting corresponding surface patches into the reference image I_c^R and the current image I_c^t for all cameras c.

The regularized position estimate $\mathbf{x}_s$ for motion refinement assumes local rigidity of the surface and tries to preserve the local structure using Laplacian coordinates, as in [Bradley et al., 2010].

The *photometric confidence* w_p is the sum of the matching errors for the neighboring positions on the tangent plane

$$w_p = 0.25(\xi_{x\pm dx,y} + \xi_{x,y\pm dy}). \tag{7.5}$$

The *regularized confidence* w_s employs a polynomial

$$w_s = \lambda_0^m + \lambda_1^m \xi_{x,y} + \lambda_2^m \xi_{x,y}^2. \tag{7.6}$$

For all examples in this chapter we use $\lambda^m = (0.5, 1, 8000)$.

7.1.5.2 Refinement across Frames

The per-frame refinement in the previous section operates on each frame independently, and can thus be run in parallel. The results are not guaranteed to be temporally consistent but consecutive frames will be very similar by construction. While temporal error in the motion estimate is mostly imperceptible, small differences in shape between successive frames can cause changes in surface normals, which will produce subtle but noticeable flickering in the visualization. To avoid this we do a final pass over the complete

sequence averaging the Laplacian coordinates in a [-1,+1] temporal window. This is implemented as an iterative process.

7.1.6 Acquisition Hardware

Our performance capture algorithm requires that the actor be illuminated by bright uniform light and recorded by $n \geq 2$ synchronized video cameras. We employ the 7-camera dynamic capture setup presented in Chapter 3 to capture the images. Note that our method does not require professional high-speed video cameras (as used by Fyffe et al. [Fyffe et al., 2011]), which would increase the cost of the system. The actor is illuminated using the light stage described in Chapter 3. Unlike previous approaches we do not require polarized light or complex controllable light patterns. The simple LED-lights used by Bradley et al. [Bradley et al., 2010] would also be sufficient for our technique.

7.2 Results

We validate our method by reconstructing several performances given by three different actors. Our results cover a wide range of expressiveness, they include visually realistic pore-level geometry, and they demonstrate our method's robustness to motion blur and occlusions, outperforming previous approaches.

Figure 7.4 shows a number of frames from our first actor, including an input image (first row), resulting geometry (second row), the geometry rendered with a grid pattern to show the temporal motion over time (third row), and the final result rendered using per-frame color information projected from the video images. This result, while demonstrating the expressive quality of our reconstructions, also illustrates how we can match a reference frame across multiple capture sequences — the first three expressions in Figure 7.4 come from sequences that were captured on different occasions, however they are still in full vertex correspondence. Figures 7.5 and 7.6 are two large color plates showing two different expressions of our first actor to better convey the spatial resolution of the reconstruction and accuracy of the tracking.

Reconstructed sequences from two additional actors are shown in Figure 7.7, demonstrating the versatility of our approach. Here we show several frames in chronological order, illustrating how the temporal reconstruction remains

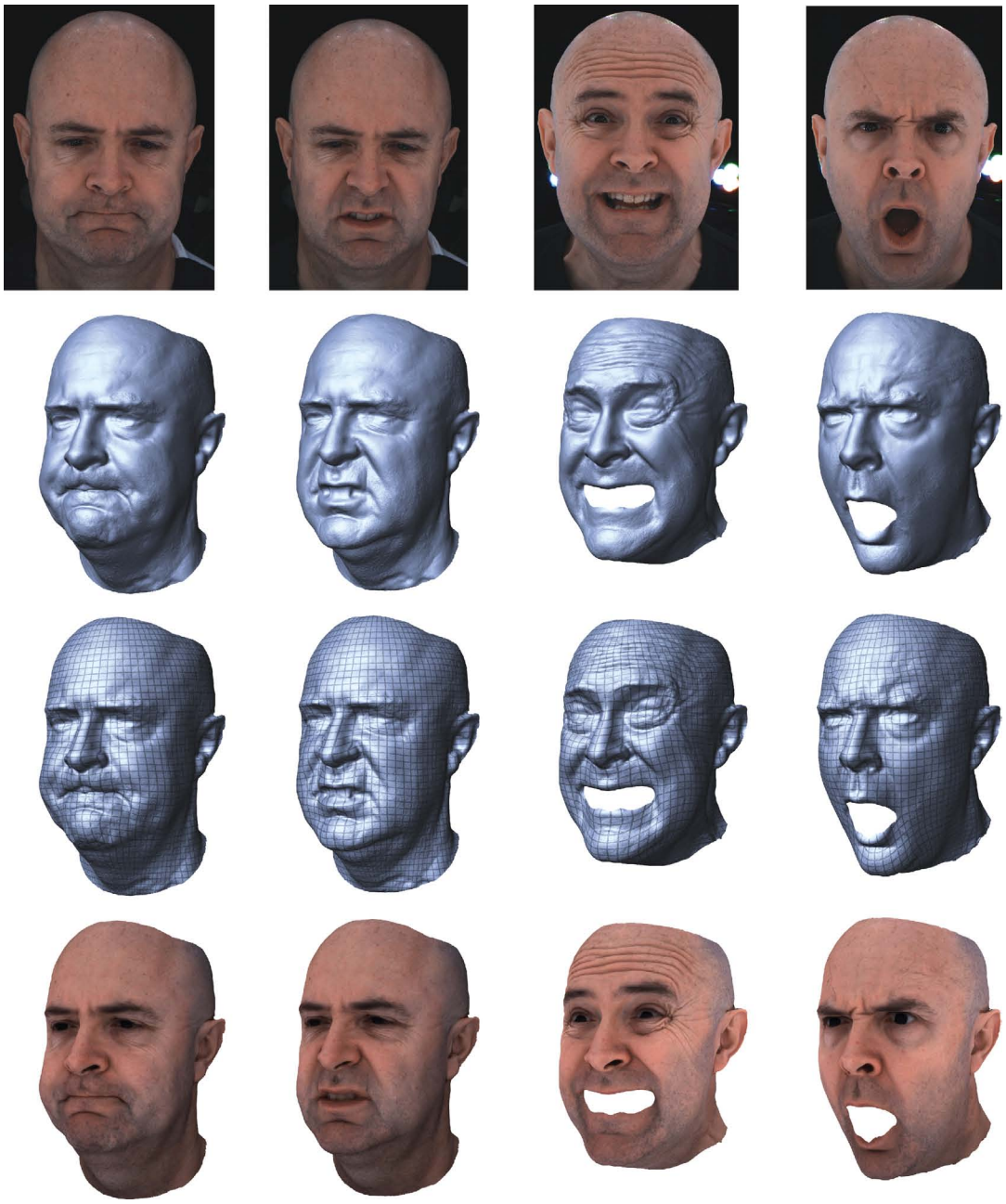


Figure 7.4: *Example frames taken from multiple different sequences of an actor. Top row: the images from one camera. Row 2: computed geometry. Row 3: geometry rendered with a grid pattern to enable a qualitative evaluation. Row 4: rendering of the computed mesh. In this figure, a single reference frame was the basis for processing multiple sequences of the actor, so the computed meshes are consistent (i.e. have corresponding mesh vertices) across all of the results.*

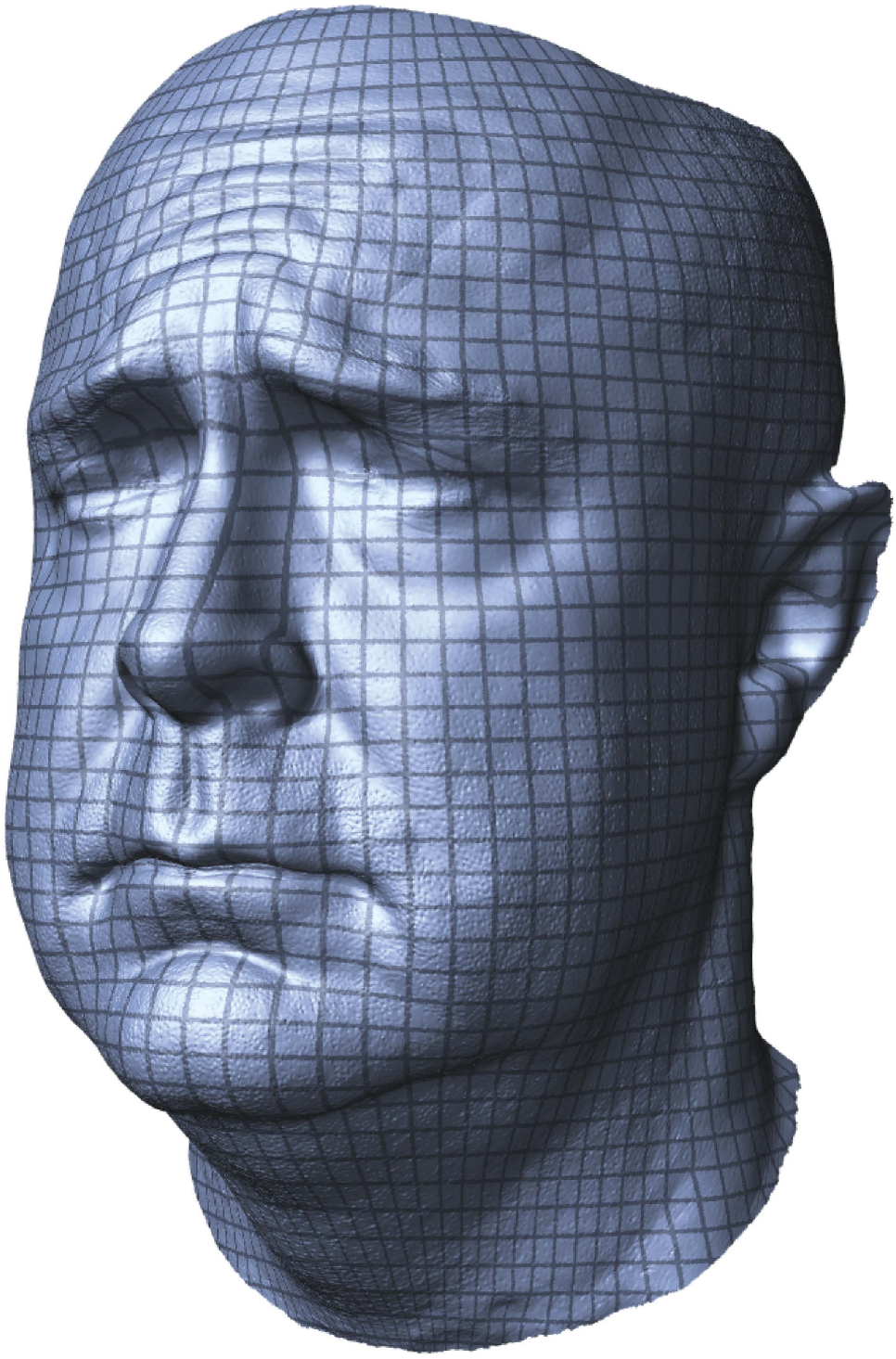


Figure 7.5: *Figures 7.5 and 7.6 show two different expressions of the same actor. The overlaid grid allows to perceive tangential motion.*

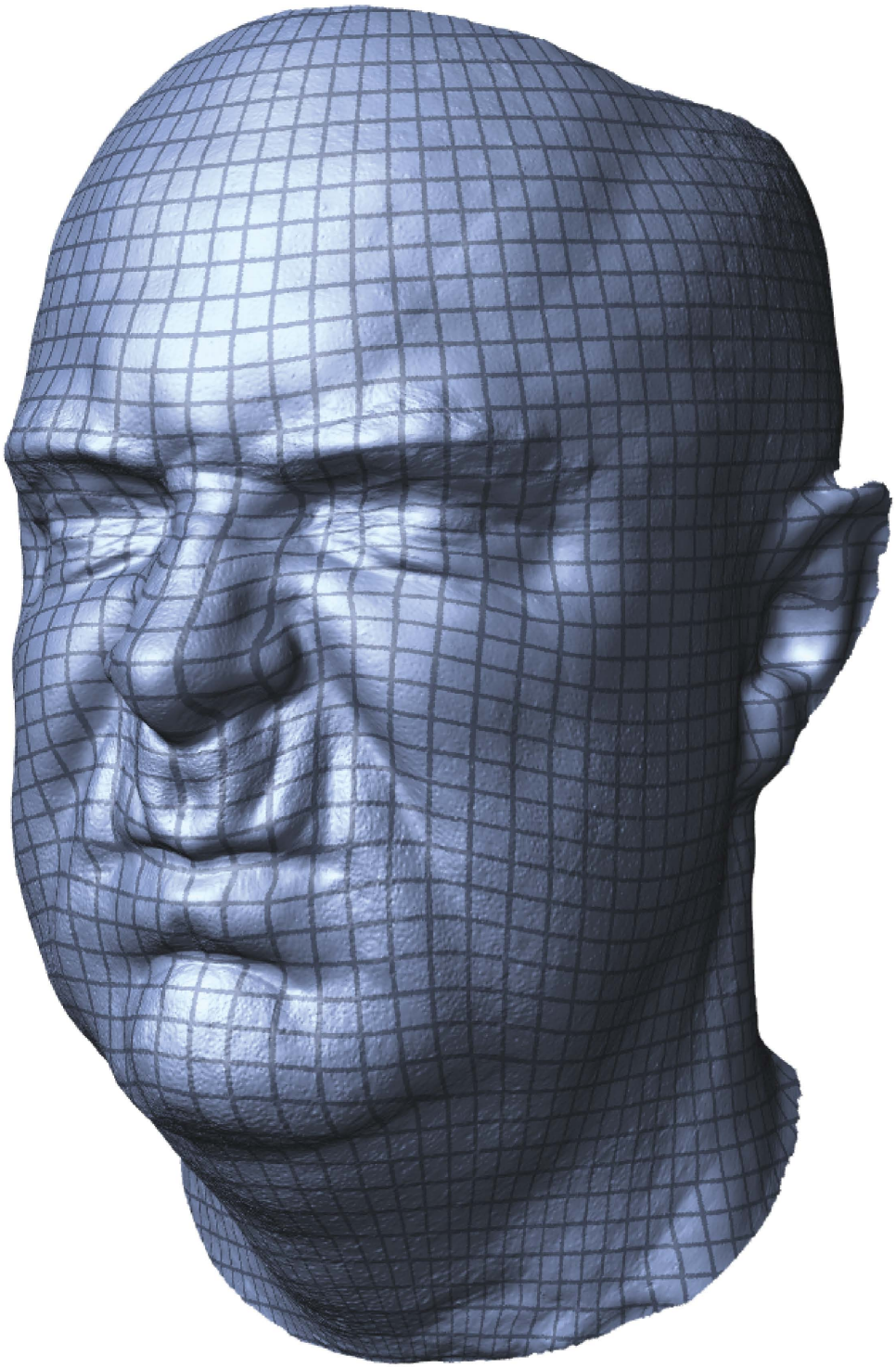


Figure 7.6: *Figures 7.5 and 7.6 show two different expressions of the same actor. The overlaid grid allows to perceive tangential motion.*

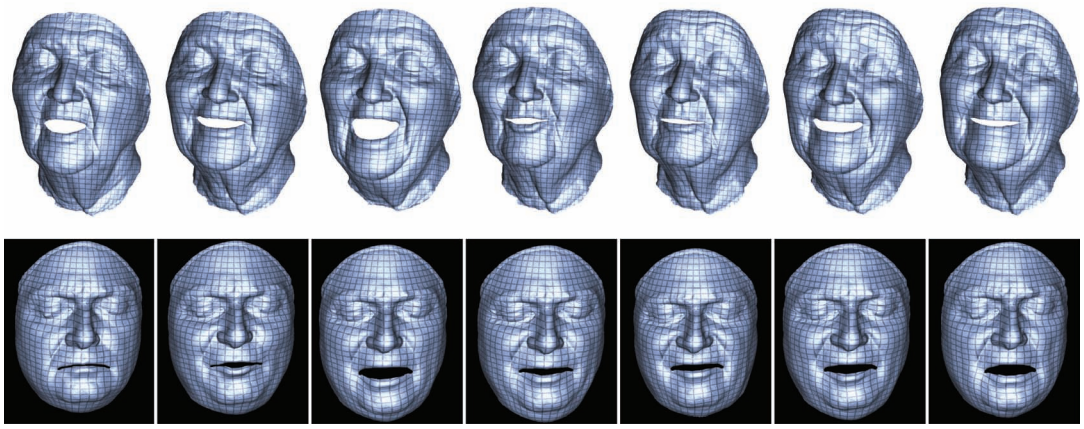


Figure 7.7: *Example on other subjects, showing frames pulled out at different time steps through a sequence. Low temporal drift is demonstrated by the consistent attachment of the superimposed grid pattern to the physical face surface.*

faithful to the true facial motion using an overlaid grid pattern. The result in the top row of Figure 7.7 particularly highlights the fine-scale spatio-temporal details that our method is able to produce.

One of the key innovations that our method relies on is our robust image-space tracking approach for deriving the temporal face motion. By dealing with error propagation directly in image space, we are able to produce more accurate motion reconstruction with less drift than techniques that rely on a potentially-distorted parameterization domain for drift correction, such as the approach of Bradley et al. [Bradley et al., 2010] (referred to just as 'Bradley et al' in the rest of this section). We illustrate this in Figure 7.8, where a short sequence of raising an actor's eyebrows to create wrinkles is reconstructed with both our approach and the method of Bradley et al. The initial geometry in both cases is computed using the system proposed in Chapter 5. Our method produces more detailed final geometry since we do not rely so heavily on spatial regularization, but also our method exhibits less drift accumulation. This can be seen in the latter three images, which show how the overlaid grid pattern deforms over time, from the first frame, to the most wrinkled frame, and then to a later frame after the wrinkles have dissipated. The temporal reconstruction of Bradley et al. is shown in yellow on the left half of the forehead, while our result is shown in blue on the right. A more regular grid pattern at the end of the sequence indicates that our approach is less susceptible to drift accumulation.

The second main contribution of our work is the application of anchor frames

to address the problems associated with sequential motion processing, such as the unavoidable tracker drift over long sequences, and complete tracking failure caused by very fast motion or occlusions. With our anchor frame reconstruction framework we can recover from such tracking failure, as we demonstrate in Figure 7.9. This result shows a sequence of lip movements in which the upper lip is occluded by the lower lip in the third image. Since tracking of the upper lip fails, the system incorrectly predicts that the motion of the upper lip drifts down onto the lower lip, indicated by the overlaid grid. Sequential tracking methods would have trouble recovering from this situation. However, due to an anchor frame later in the sequence, our method is able to successfully track the upper lip backwards from the anchor frame to the occluded frame, automatically restoring tracking after the occlusion.

The combination of robust image space tracking and anchor frames allows us to successfully reconstruct very fast motions, even those containing motion blur. We demonstrate this in Figure 7.10, which contains a short sequence of an actor opening his mouth very quickly, and we compare our result again to the method of Bradley et al. Our method is able to produce a more accurate reconstruction.

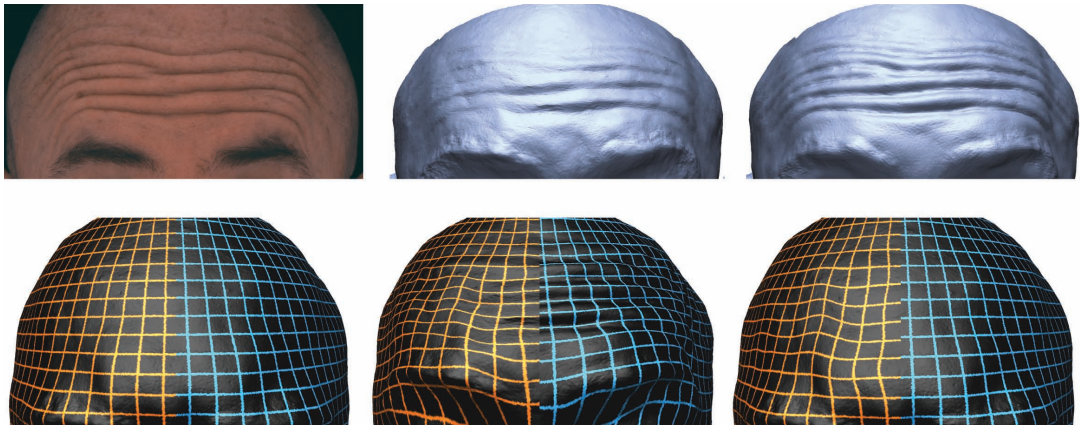


Figure 7.8: Comparison with the method of Bradley et al. - *The top row demonstrates differences in the reconstructed geometry. From left to right: a zoom onto the forehead in an input image, 3D reconstruction of Bradley et al, our 3D reconstruction showing improved fidelity. The images on the bottom row show results for Bradley et al on the left-side of the head and our results on the right-side, for three time steps starting with unwrinkled forehead, then wrinkled forehead, then back to unwrinkled. Changes in the attachment of the superimposed grid to the face between the first and last images demonstrate drift in the tracking, with our method showing significantly less drift.*

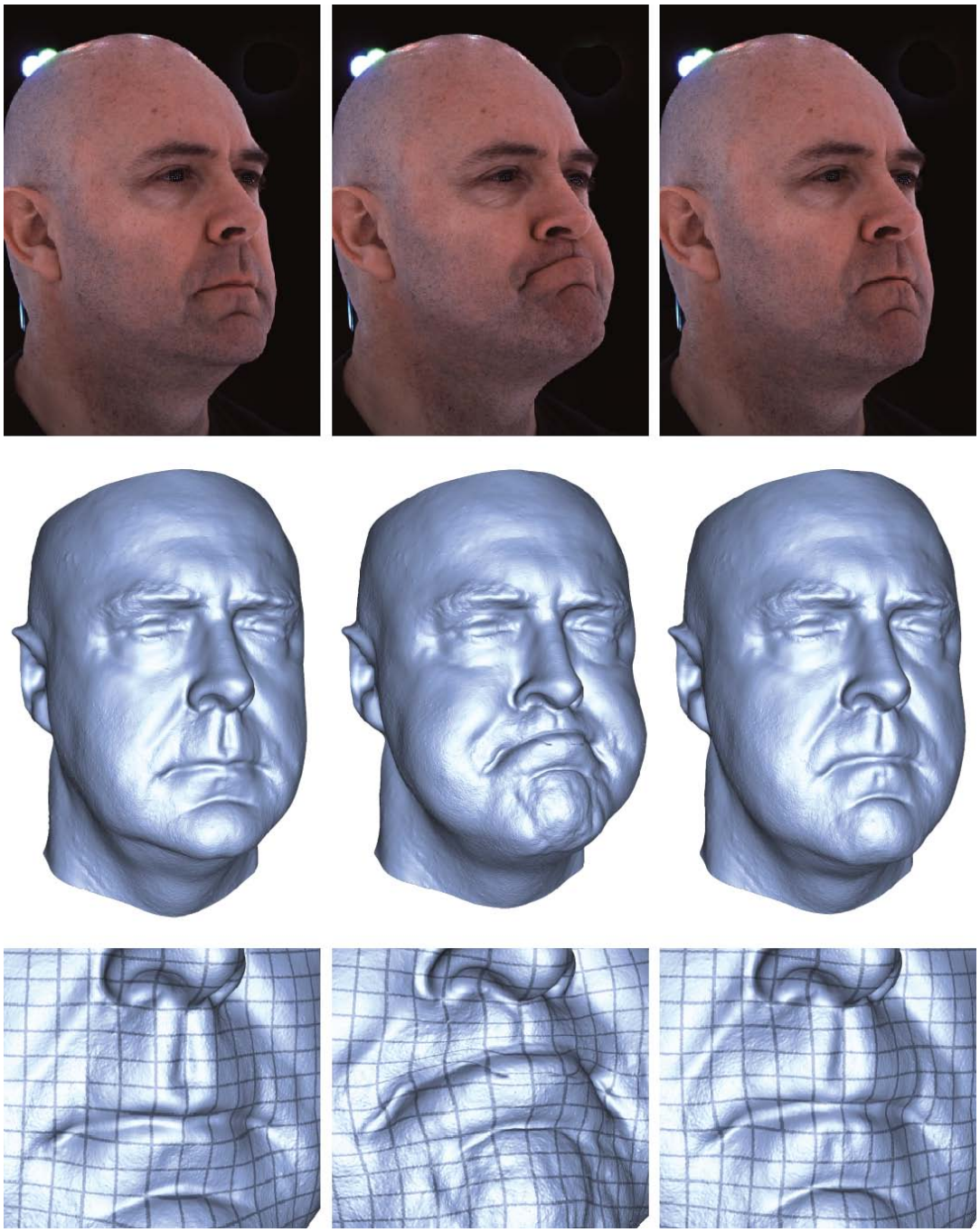


Figure 7.9: *Several frames from a sequence in which there is significant occlusion and reappearance of structure around the mouth. Our method of anchor frames combats this by partitioning the sequence into clips, and doing image-space tracking from the start and end of each clip.*

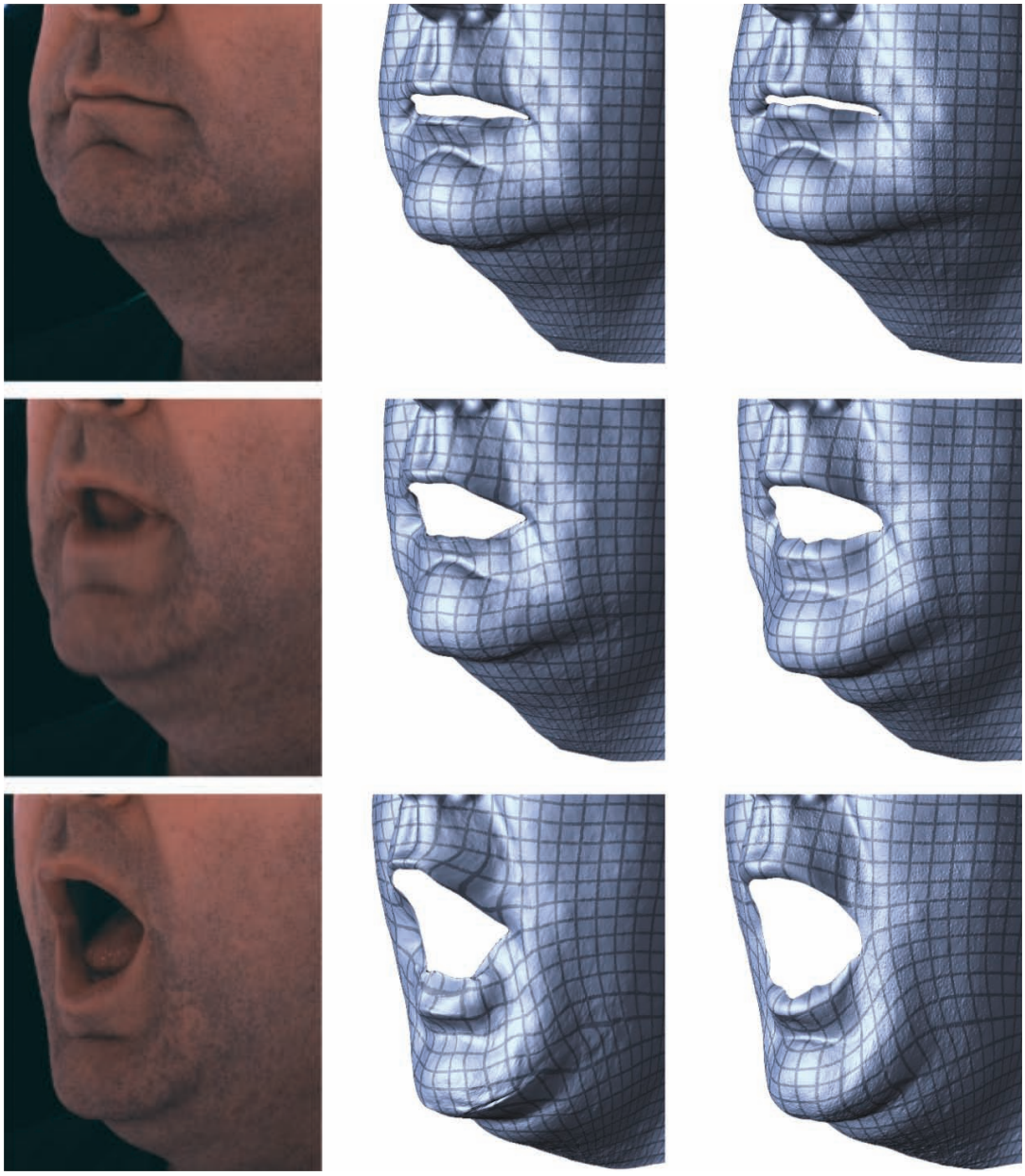


Figure 7.10: *Left: a zoom onto the mouth for several frames in a sequence in which there is fast facial motion and motion blur in the images. Center: the reconstruction of Bradley et al. Right: our reconstruction, with greater fidelity around the mouth.*

Analysis Here we assess the behavior of our algorithm under varying numbers of anchor frames. This assessment, depicted in Figure 7.11, demonstrates that our algorithm is relatively insensitive to the number of anchor frames, when present in typical amounts. For this analysis we ran our tracking method 9 times on the same sequence, using the same reference frame but varying the anchor frames. The sequence consists of 400 frames and contains a number of expressive as well as neutral poses, as illustrated by the images at the bottom of the figure. In 4 of the executions the anchor frames are manually selected (with 1, 3, 5 and 8 anchors), and in the other executions they are automatically selected based on the matching error (which is graphed in the background in purple); these 5 executions chose anchor frames that had the highest 10, 25, 50, 75 and 100 percentile matching error. At the low end, with only 1 anchor frame, we expect drift to accumulate since tracking is sequential (as in previous methods). At the high end, with 100% anchors, the method should degenerate completely because there is no frame-to-frame tracking at all. However, with a reasonable number of anchor frames we expect the results to be stable. We have no ground truth for measuring the difference between executions, so we chose the result with 8 anchors as the baseline for comparison. These anchor frames approximately partition the sequence into individual expressions bounded by neutral poses, which is exactly the situation where we expect our method to perform best. For each other result sequence, we measure the average image-space tracking error for each frame in pixels, compared to the baseline result, and accumulate the error across the sequence. This accumulated error is shown in the horizontal bar chart on the right side of the figure. As expected, using only 1 anchor frame produces significant error due to drift accumulation. When the number of anchor frames is very high the error is also large because many frames do not match well to the reference frame directly. However, using anywhere from 3 anchors to 10% of the frames as anchors produces similar results, indicating that our algorithm is relatively insensitive to the number of anchor frames.

Our initial presentation of the concept of anchoring involved a single reference frame. It may happen that a given reference frame does not yield a good distribution of anchor frames in the whole sequence, so that the benefits of anchor-based reconstruction are lost in some places. In this case, it is not necessary to maintain the same reference frame for the entire sequence. A subset of the sequence can be matched to one reference frame, followed by a change of the reference frame to one of the processed frames, if the switch would yield a better anchor frame distribution for the remainder of the sequence. The result shown in the latter three frames of Figure 7.4 is generated in this way, where the reference frame starts off as a neutral expression and

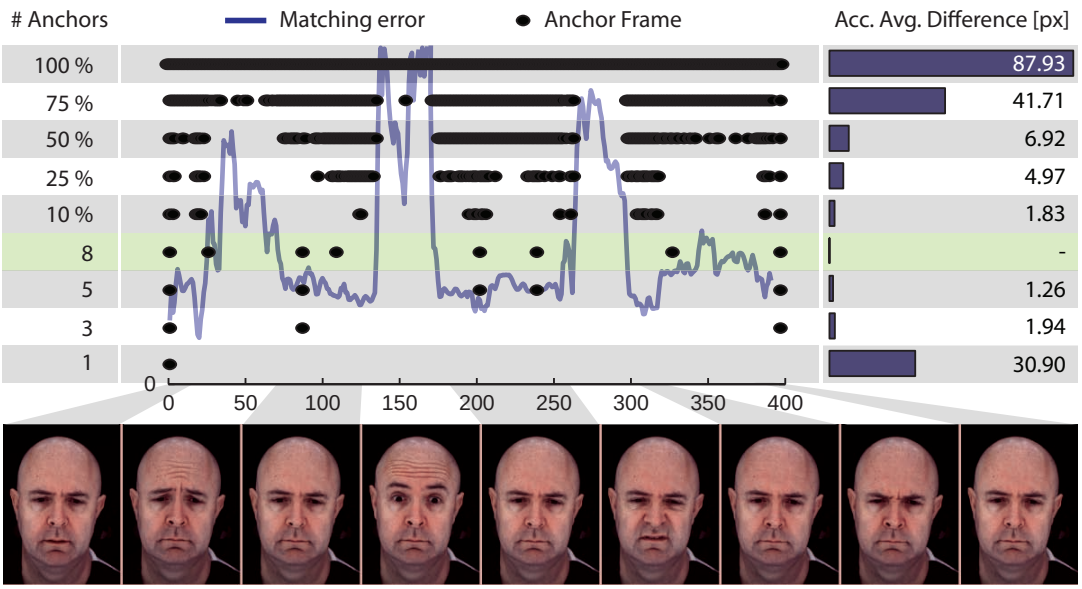


Figure 7.11: Anchor frame analysis - *Analysis of quantity and placement of anchor frames. The tracking error increases both when too many and when too few anchor frames are selected.*

then switches to a pose with the mouth open, since the mouth remains open for most of the sequence. The two reference frames would need to be in full vertex correspondence, but by switching the reference to a frame that has already been processed, the correspondence between the two frames is directly available.

7.3 Discussion

This chapter presents a performance capture algorithm that can acquire expressive facial performances with visually-realistic pore-level geometric details. Here we discuss some of the research opportunities for extending the system.

In common with previous methods, we do not expect to obtain a faithful 3D reconstruction of the eye geometry or facial hair, since these tend to violate stereo and temporal brightness constancy assumptions in image space. The brightness constancy assumption is also violated for skin that undergoes strong wrinkling, which may cause problems for optical flow computation. In Chapter 8 we will introduce a complementary method to remedy these

issues. Visual artifacts may also be seen at the mesh boundaries, however these can easily be cleaned in a post-process.

Our image-space tracking requires a stereo deployment where all cameras capture the full face. This contrasts with a stereo deployment like that of Bradley et al [Bradley et al., 2010], where the cameras are optically zoomed to capture small patches of the face, and a single point on the face often migrates between stereo views during a sequence[1]. Our approach could be extended to this situation if we were to combine the stereo images into one image, for example using the "unwrap mosaics" method of Rav-Acha et al. [Rav-Acha et al., 2008].

The reference frame is matched to the anchor frames in image space, as described in Section 7.1.2. A more flexible approach, and a straightforward extension, would be to do the matching in 3D via the meshes that are computed in Stage 1 of the pipeline. Thus matching would succeed whenever facial expression is the same in the reference and anchor frames, and would no longer require a similar orientation of the head relative to the cameras.

Anchoring can be a powerful tool for integrating multiple face performances of an actor over an extended period. As shown in the results section, a reference frame can be taken from one sequence but used to generate anchor frames in another sequence. This provides a way to propagate a single mesh across different capture sessions for an actor (including the case where the camera positions or calibration may have changed somewhat between the sessions), and to embed the full corpus of facial performance data for the actor into a single coordinate frame.

An extension of the work here would be to use multiple reference frames simultaneously. For example, facial performance capture could be applied to a sequence in which an actor adapts a set of FACS poses [Ekman and Friesen, 1978] with careful supervision to yield best possible results. The frames with the FACS poses could then be used as a set of high quality reference frames that could be used simultaneously when processing subsequent sequences (because the meshes have consistent triangulation).

Finally we believe that there is an interesting new avenue for research in segmenting the face, and applying the concept of anchored reconstruction at the level of individual parts of the face. This relates to a contribution of our work, which was to show how a long image sequence can be decomposed into anchored clips for 3D reconstruction. Face segmentation will provide

[1] For this reason we are not able to reconstruct the original datasets of Bradley et al. Instead we ran their tracking algorithm on our capture data for the comparison.

an orthogonal way of decomposing the process, and we plan to explore this extension.

7.4 Conclusion

In this chapter we presented a new passive technique for high-quality facial performance capture, based on two key technical innovations. First, we employ a robust tracking algorithm that integrates all of the pixel tracking in image space and uses the integrated result to propagate a single reference mesh to each target frame in parallel. Second, leveraging the fact that facial performances tend to contain repetitive motions, we introduce "anchor frames" defined as those where the facial expression is similar to the reference frame. After locating the anchor frames automatically, we compute pixel tracking directly from the reference frame to anchor frames. By using the anchor frames to partition the sequence into clips and independently matching clips, we are able to bound tracker drift, correctly handle occlusion and motion blur, and process capture sequences in parallel. We can even match frames between multiple capture sessions recorded on different occasions, yielding a single deformable mesh that corresponds to every performance an actor gives.

Our method produces detailed 3D geometry in full temporal correspondence, even for the most expressive of performances undergoing very fast motion, without the requirement of hand-placed markers or face makeup. We have demonstrated our technique on a number of example performances given by different actors, and have also shown how our anchored-reconstruction approach combined with our robust image-space tracking method can yield more accurate results than a current state-of-the-art technique [Bradley et al., 2010], particularly in the presence of motion blur and highly expressive wrinkles where drift tends to accumulate faster.

To our knowledge, ours is the first method to passively reconstruct 3D facial performances with visually realistic pore-level geometric details, while demonstrating robustness to fast motions. A system of this type would be very useful for facial animation applications, such as performance transfer from one actor to another, in particular given the high-resolution geometry and expressive motions our method is able to reconstruct.

CHAPTER 8

Cancelling Ambient Occlusion

The performance capture algorithm presented in Chapter 7 employs optical flow to densely track a facial performance over time. During such a facial performance, the facial skin deforms, which induces a change in shading. This time-varying shading poses problems for optical flow algorithms, which typically rely on image brightness constancy. In this chapter we present a general technique for improving space-time reconstructions of deforming surfaces, which are captured in an video-based reconstruction scenario. The approach simultaneously improves both the acquired shape as well as the tracked motion of the deforming surface. The method is based on factoring out surface shading, computed by a fast approximation to global illumination called ambient occlusion. This allows us to improve the performance of optical flow tracking that mainly relies on constancy of image features, such as intensity. While cancelling the local shading, we also optimize the surface shape to minimize the residual between the ambient occlusion of the 3D geometry and that of the image, yielding more accurate surface details in the reconstruction. The enhancement is independent of the actual space-time reconstruction algorithm. We experimentally measure the quantitative improvements produced by the algorithm using a synthetic example of deforming skin, where ground truth shape and motion is available, and demonstrate improved performance for several well-known optical flow algorithms. We

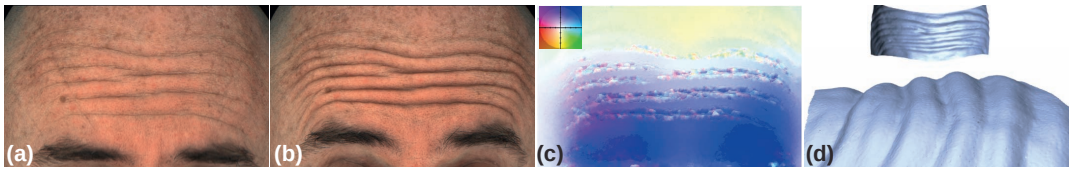


Figure 8.1: Problems with deforming skin - *Motion and geometry reconstruction problems for deforming skin. Two video images (a) and (b) showing a skin wrinkle forming. (c) shows that a state-of-the-art optical flow algorithm [Brox et al., 2004] fails for such illumination changes (flow vectors are visualized using the color code shown in the inset). (d) shows that the reconstructed geometry is overly smooth, as the reconstruction algorithm has insufficient photometric information in this area.*

further demonstrate the enhancement on a real-world sequence of human face reconstruction.

8.1 Problem Definition and Method Overview

To reconstruct a deforming surface, we must capture both the time-varying shape of the surface as well as its motion, tracking each point over time. In recent approaches, this has been achieved by combining multi-view reconstruction techniques with image-based optical flow tracking [De Aguiar et al., 2007, Bradley et al., 2010, Beeler et al., 2011]. As we mentioned in the introduction, if the surface contains local high-frequency deformations then both the optical flow tracking and the reconstructed surface shape can be inaccurate. Figure 8.1 highlights these problems for a real-world reconstruction example of deforming skin from Chapter 7. Here we see that the local shading variation causes incorrect flow vectors for a well-known optical flow technique [Brox et al., 2004]. Furthermore, the reconstructed surface geometry is overly smooth in the wrinkle regions, as the reconstruction algorithm has insufficient photometric information in this area. In order to compensate for these problems, the approach proposed in this chapter is to compute and factor out local surface shading, approximated by ambient occlusion.

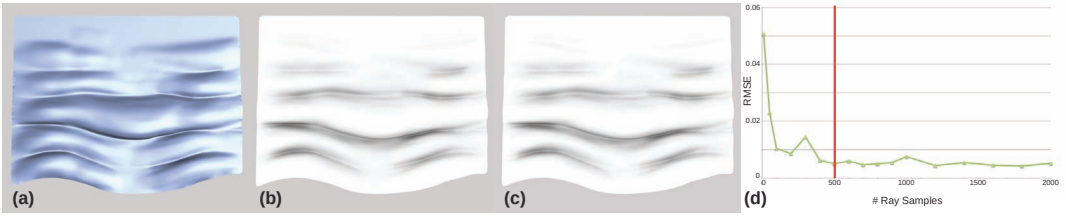


Figure 8.2: Ambient occlusion computation - *(a) 3D surface patch. (b) Ground-truth computed using Monte-Carlo ray-tracing with 100,000 ray samples. (c) The proposed approximation with 500 samples is almost identical. (d) RMS error of the proposed approximation for different sample sizes.*

8.1.1 Ambient Occlusion

Ambient occlusion is a global shading method that approximates global illumination [Zhukov et al., 1998]. It does not take into account effects such as cast shadows, inter-reflections or subsurface scattering. However, in a setting with diffuse or omnidirectional illumination, ambient occlusion approximates global illumination well. Ambient occlusion is defined as

$$A(\mathbf{x}) = \frac{1}{\pi} \int_{\Omega} V(\mathbf{x}, \omega) \langle \mathbf{n}(\mathbf{x}), \omega \rangle \, d\omega, \tag{8.1}$$

where $\mathbf{x}$ is a point on the surface, $\mathbf{n}(\mathbf{x})$ the normal at this point, $\langle \cdot, \cdot \rangle$ denotes the inner product and $V(\cdot, \cdot)$ is a visibility function that is 0 if the ray ω is occluded and 1 otherwise. The integral is formed over the hemisphere Ω, which makes ambient occlusion costly to compute in general, especially for large meshes. Several methods have been proposed for efficient ambient occlusion approximations (see [Méndez-Feliu and Sbert, 2009] for a survey).

In this chapter, we will refine the shape of a surface based on the computed ambient occlusion, and so the quality of the refinement depends on the accuracy of the ambient occlusion estimation. At the same time, the refinement method is iterative, and so we aim for fast computation. To meet these requirements, we implemented a fast ray-tracing approach with deterministic ray samples. We use the Intel Embree high-performance ray-tracing library[1], which is designed exactly for this purpose. However, instead of traditional Monte-Carlo stochastic ray tracing, we use a deterministic cosine-distribution of samples around each vertex normal. While this approach in-

[1] http://software.intel.com/en-us/articles/embree-photo-realistic-ray-tracing-kernels/ (accessed 27. June 2012).

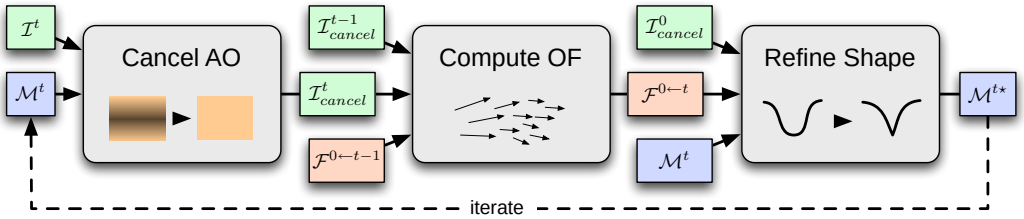


Figure 8.3: Overview of the algorithm - *The algorithm consists of three major stages. Stage 1 computes ambient occlusion on the mesh $\mathcal{M}^t$ and removes it from the input image I^t. This improves the estimation of the flow field $F^{0\leftarrow t}$ in Stage 2. Stage 3 produces a refined shape $\mathcal{M}^{t\star}$ by minimizing the residual of the observed shading and computed ambient occlusion.*

troduces a small amount of bias in the result, the spatially-varying noise in ambient occlusion is greatly reduced for the same number of samples, allowing us to compute a close approximation in a matter of seconds rather than tens of minutes with the Monte-Carlo approach. Figure 8.2 shows the ambient occlusion computation for a 3D surface patch. In order to determine how many ray samples to use, we plot the RMS error for different sample sizes (Figure 8.2 (d)) compared to ground truth ambient occlusion computed using Monte-Carlo ray-tracing with 100,000 samples (Figure 8.2 (b)). We found that any quality gain beyond 500 samples (Figure 8.2 (c)) was negligible. Note that the deterministic approximation does not converge exactly to the ground truth, as we see in the RMS error plot, and the convergence is not a monotonically decreasing function. However, the result is visually almost identical to the ground truth, and in practice we found that inaccuracies of such a small magnitude had no effect on the algorithm.

8.1.2 Method Overview

We now give an overview of the proposed technique for improving reconstructions of deforming surfaces by cancelling ambient occlusion. A pictorial representation of the method can be found in Figure 8.3. Given a sequence of reconstructed meshes and the corresponding calibrated camera images, the algorithm processes the frames sequentially with three main steps per frame:

1. **Cancel Ambient Occlusion** - The ambient occlusion of the surface is computed and projected onto each image plane, then divided out of the image. Ambient occlusion is computed as described in Section 8.1.1.

2. **Motion Improvement** - Optical flow is computed on the shading-free images created by cancelling ambient occlusion (Section 8.2).
3. **Shape Improvement** - The 3D shape is refined to minimize the ambient occlusion residual when cancelling from the images, leading to a surface that better captures fine details such as wrinkles (Section 8.3).

These steps are iterated until the shape and motion refinement becomes negligible. In our experiments, typically only two to three iterations are required.

The presented algorithm is completely independent of the original 3D reconstruction method and the optical flow algorithm. In Section 8.4 we will show improvements to the facial geometry reconstruction method introduced in Chapter 5. We also show that several common optical flow methods [Lucas and Kanade, 1981, Horn, 1981, Brox et al., 2004, Zimmer et al., 2011, Werlberger et al., 2010] perform much better on the ambient occlusion cancelled images than the original images.

8.1.3 Notation

The following notation will be used throughout this chapter.

$\mathcal{M}^t(\mathbf{x})$ is a mesh at frame t, defined over vertices $\mathbf{x}$. $\mathcal{M}^t$ will be used for short.

$I^t(\mathbf{p})$ is an image at frame t, defined over pixels $\mathbf{p}$. I^t will be used for short.

$F^{t-1\leftarrow t}(\mathbf{p})$ is the flow from I^t to I^{t-1}. $F^{t-1\leftarrow t}$ will be used for short.

8.2 Motion Improvement

The flow field $F^{t-1\leftarrow t}$ is improved by removing shading caused by ambient occlusion of the reconstructed meshes $\mathcal{M}^{t-1}$ and $\mathcal{M}^t$ from the images I^{t-1} and I^t, respectively. The shading is removed via

$$I^t_{cancel} = \frac{I^t}{P\left(A(\mathcal{M}^t)\right)}, \tag{8.2}$$

where $P(\cdot)$ projects the ambient occlusion computed for the mesh $\mathcal{M}$ onto the image I. The improved flow field $F^{t-1\leftarrow t}_{cancel}$ is integrated with $F^{0\leftarrow t-1}_{cancel}$ to produce the motion estimation $F^{0\leftarrow t}_{cancel}$ from frame t to the first frame. The flow is estimated backwards to facilitate easy warping of the first frame to frame t, which will be required in the next stage of the algorithm.

8.3 Shape Improvement

The motivation to refine the shape stems from the observation that in the assumed setting most shading changes are caused by shape deformation. The shape is refined such that the predicted shading corresponds to the observed shading. The shape improvement consists of the following steps:

1. Compute the observed shading $A'(\mathbf{x})$ from the images.
2. Compute the ambient occlusion $A(\mathbf{x})$ on the surface.
3. Compute the refinement $\delta(\mathbf{x})$ based on $A'(\mathbf{x})$ and $A(\mathbf{x})$.
4. Update vertex positions $\mathbf{x}^\star = \mathbf{x} + \delta(\mathbf{x})\mathbf{n}(\mathbf{x})$.

These steps are performed iteratively for all vertices of a mesh. Note that vertices are displaced only along the normal direction. Constraining the refinement to a single dimension greatly reduces computational complexity and increases robustness of the algorithm. If the surface contains many high-frequency details, we found that a low-pass filter of the normals produces better results. Although normal vectors are updated in each iteration in order to compute accurate ambient occlusion, the displacement directions of the vertices remain constant. The steps of the algorithm are explained in more detail in the following.

The observed shading is computed from the input images. For a single image, $A'(\mathbf{x})$ is computed as

$$A'(\mathbf{x}) = \frac{I^t(\mathbf{q})}{W^{0\to t}(I^0_{cancel}(\mathbf{q}))}, \tag{8.3}$$

where $\mathbf{q}$ is the projection of $\mathbf{x}$ onto the image plane, and $W^{a\to b}$ is a warping function that warps an image from frame a to frame b given the flow field $F^{a\leftarrow b}_{cancel}$ computed in Section 8.2. If multiple views exist, $A'(\mathbf{x})$ can be computed as a (weighted) average from all images.

The predicted shading $A(\mathbf{x})$ is computed from the mesh $\mathcal{M}^t$ using ambient occlusion, as described in Section 8.1.1. The refined position $\mathbf{x}^\star$ for a vertex $\mathbf{x}$ of the mesh $\mathcal{M}^t$ is computed as

$$\mathbf{x}^\star = \mathbf{x} + \delta(\mathbf{x})\mathbf{n}(\mathbf{x}). \tag{8.4}$$

Using the residual $\delta_A(\mathbf{x}) = s(A(\mathbf{x}) - A'(\mathbf{x}))$, where s matches the scale of the geometry, the refinement is computed as

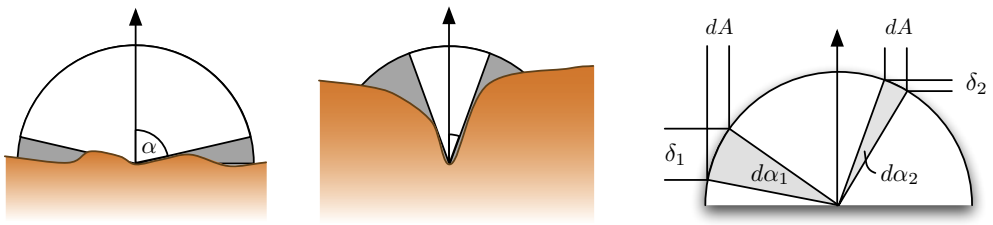


Figure 8.4: Non-linear regularization - *This figure depicts the non-linear dependency of the displacement on the observed shading in 2D. The left and middle drawings show the relation of the half-angle α to the ambient occlusion for two different cases. The right figure illustrates that the same perturbation dA in ambient occlusion leads to different $d\alpha$ and therefore different displacements δ depending on the concavity of the surface.*

$$\delta(\mathbf{x}) = \frac{\gamma(A'(\mathbf{x}))\delta_A(\mathbf{x}) + \lambda\delta_L(\mathbf{x})}{\gamma(A'(\mathbf{x})) + \lambda}, \tag{8.5}$$

where λ is a parameter that controls the influence of the regularization. We use $\lambda = 2$. The regularized offset $\delta_L(\cdot)$ is computed using Laplacian coordinates as

$$\delta_L(\mathbf{x}) = \langle \nabla^2 \mathcal{M}(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle - \eta(\mathbf{x}), \tag{8.6}$$

where $\eta(\cdot)$ controls the target shape. The default choice is $\eta(\mathbf{x}) = 0$ for all vertices, which prefers smooth solutions. If the shape of the input meshes can be considered mostly accurate, then setting $\eta(\cdot)$ to the Laplacian coordinates of the input mesh is a better choice. In these cases the regularization will try to maintain the input shape.

The non-linear function $\gamma(\cdot)$ controls the refinement strength depending on the observed shading $A'(\mathbf{x})$. This function accounts for the non-linear influence of noise in $A'(\mathbf{x})$ on the shape. The same perturbation of $A'(\mathbf{x})$ would induce larger perturbation of the shape in areas of lower concavity, as depicted in Figure 8.4.

From the illustration in Figure 8.4, an ambient occlusion value can be characterized by a half-angle α, defining a cone of visibility.

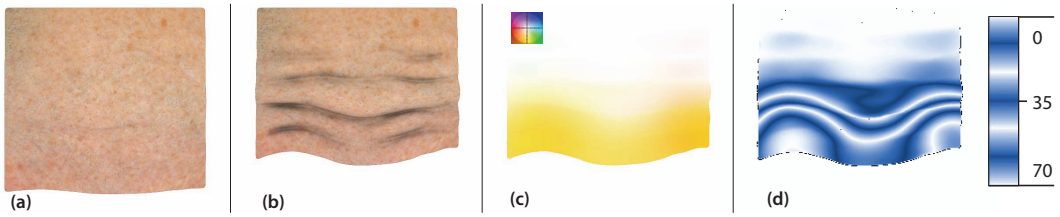


Figure 8.5: Synthetic sequence - *Figure (a) and (b) show two frames of the synthetic sequence used in the quantitative evaluation. (c) shows the ground truth flow field in the Middlebury color scheme. To better visualize the large variation in displacement (0px-70px) we employ the iso-contour scheme depicted in (d).*

Then $\gamma(\cdot)$ can be written as a function of α,

$$\gamma(\alpha) = \frac{\cos(\alpha) + \epsilon}{1 + \epsilon}, \tag{8.7}$$

where ϵ is a small parameter that controls the lower bound of $\gamma(\cdot)$. Setting ϵ to 0 will prevent refinement in planar and convex areas. We use $\epsilon = 0.1$. The observed shading $A'(\mathbf{x})$ is related to $\gamma(\cdot)$ via the angle α through

$$\begin{aligned} A'(\mathbf{x}) &\approx \frac{1}{\pi} \int_{\Omega} V(\mathbf{x}, \omega) \langle \mathbf{n}(\mathbf{x}), \omega \rangle \, d\omega & (8.8) \\ &= \frac{1}{\pi} \int_{0}^{2\pi} \int_{0}^{\alpha} \cos(\phi) \sin(\phi) \, d\phi d\theta & (8.9) \\ &= \sin^2(\alpha) & (8.10) \end{aligned}$$

and thus

$$\gamma(A'(\mathbf{x})) = \frac{\sqrt{1 - A'(\mathbf{x})} + \epsilon}{1 + \epsilon}. \tag{8.11}$$

Refining convex areas less than concave ones is also very valuable when assuming the input shape is mostly accurate, as areas that are well visible are more likely to be of correct shape than concave ones.

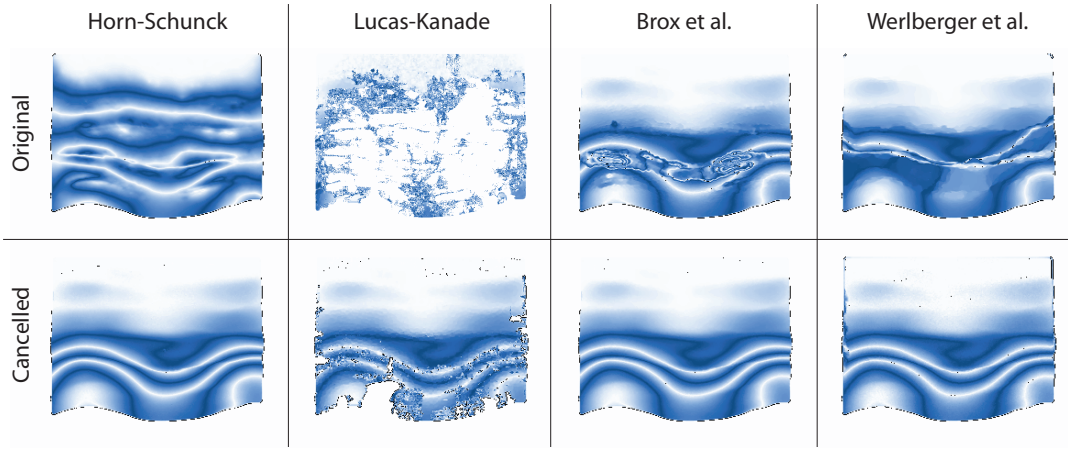


Figure 8.6: *This figure shows computed flow fields on the original (top row) and cancelled images (bottom row) for four well known algorithms. All algorithms have problems in areas where their core matching assumptions are violated due to the change in shading. Cancelling ambient occlusion increases the performance of all algorithms substantially. The ground truth flow field is shown in Figure 8.5 and the computed errors are listed in Table 8.1.*

8.4 Results

In this section we present the results of the proposed algorithm, starting with an evaluation using a synthetic dataset, and then demonstrating the improvement we achieve on a real-world capture sequence.

8.4.1 Quantitative Evaluation

To quantitatively measure the effect of the presented optical flow improvement method, we designed a synthetic test sequence giving us ground truth motion. The sequence consists of a skin-textured surface patch that undergoes wrinkling while deforming. The deformation was created in Maya using blend shapes. We evaluate six different optical flow algorithms: pyramid implementations of Lucas-Kanade [Lucas and Kanade, 1981] and Horn-Schunck [Horn, 1981], as well as the Horn-Schunck algorithm with added gradient constancy, the method of Brox et al. [Brox et al., 2004], Zimmer et al. [Zimmer et al., 2011], and Werlberger et al. [Werlberger et al., 2010].

Algorithm	Endpoint Error (EE) [px]		Upper 33% EE [px]	
	Original	Refined	Original	Refined
Horn-Schunck	13.86±12.06	0.15±0.16	28.79±5.85	0.31±0.19
Horn-Schunck (g.)	3.27±7.83	0.15±0.18	9.86±11.34	0.32±0.22
Brox et al.	1.77±6.03	0.13±0.14	10.72±22.22	0.29±0.16
Zimmer et al.	2.53±6.89	0.15±0.20	8.38±12.21	0.32±0.28
Werlberger et al.	3.14±7.07	0.33±1.14	9.55±9.82	0.76±1.95
Lucas-Kanade	3.02±8.34	1.38±5.59	278.3±85.7	24.1±52.8

Table 8.1: *The Table summarizes mean Endpoint Errors (EE) and standard deviations for all benchmarked algorithms. The proposed method greatly improves the performance of all algorithms. The errors reported for Lucas and Kanade are less indicative because they include outliers that are not caused by wrinkling and the completeness of the result differ substantially (25% on the original and 87% on the cancelled sequence).*

We used the openCV[2] implementation for Lucas-Kanade, the flowLib[3] library for Werlberger et al. and the implementations of the other algorithms were kindly provided by Zimmer et al. [Zimmer et al., 2011]. Figure 8.5 shows the two frames that were used for the benchmark, as well as the backward flow field. The flow visualization from the Middlebury evaluation [Baker et al., 2011] is not very meaningful in this case (Figure 8.5 (c)) since the motion is primarily 1-dimensional, so we use a custom visualization based on the flow magnitude for this evaluation (Figure 8.5 (d)).

In this section we chose two temporally distant frames on purpose, to better visualize the impact of the method. The flow algorithms are also improved for closer frames as shown in Section 8.4.2 but the effect is, of course, smaller if there is less change in shading. A selection of the evaluation benchmark results are shown in Figure 8.6, the full results are listed in Table 8.1 and evaluated in more depth in Section 8.4.2. We report the Endpoint Error [Baker et al., 2011] for the complete patch as well as the Endpoint Error for the worst 33% in order to account for the fact that only part of the image exhibits shading change.

As can be seen from Figure 8.6, the flow methods have problems estimating the motion within the wrinkles due to shading change. Cancelling the ambient occlusion gives consistently better results for all algorithms. Algorithms

[2] http://sourceforge.net/projects/opencvlibrary (accessed 27 June 2012).

[3] http://gpu4vision.icg.tugraz.at/index.php?content=subsites/flowlib/flowlib.php (accessed 27 June 2012).

Algorithm	Parameters	Original	Refined
Horn-Schunck	α	1500	10
Horn-Schunck (g.)	α/γ	1500/200	150/5
Brox et al.	α/γ	20/5	20/0
Zimmer et al.	α/γ	75/20	75/5
Werlberger et al.	$\alpha/\beta/\lambda/\epsilon$	5/1.0/30/0.001	5/0.5/50/1.5
Lucas-Kanade	window/iters.	5/100	5/100

Table 8.2: *Parameters of the benchmarked algorithms.*

that assume only brightness constancy benefit most, but also algorithms that include gradient constancy show better performance. On the cancelled sequence, the gradient constancy assumption is not that influential anymore and, for example, Brox et al. produce the lowest Endpoint Error without gradient constancy. A special case is Lucas-Kanade. While its performance is greatly improved (see the second column of Figure 8.6) the reported errors are less indicative because they include outliers that are not caused by wrinkling and the completeness of the result differs substantially (25% on the original and 87% on the cancelled sequence).

The parameters were empirically chosen to produce the best result that can be achieved before and after cancelling ambient occlusion. The used parameters are listed in Table 8.2. In general, we found that parameter tuning for the original images was more challenging since the resulting flow was very sensitive to parameter changes, where on the cancelled images flow computation was more robust to parameter variations. To achieve decent flow estimations for the original images we had to choose high regularization parameters, while for the cancelled images significantly lower regularization produced the most accurate results.

8.4.2 Extended Evaluation of Results

The following figures present an extended evaluation of several well-known optical flow algorithms; *Horn and Schunck* (Figure 8.7), *Horn and Schunck with Gradient Constancy* (Figure 8.8), *Brox et al.* (Figure 8.9), *Zimmer et al.* (Figure 8.10), *Werlberger et al.* (Figure 8.11) and *Lucas and Kanade* (Figure 8.12). Figure 8.13 shows the original images and the result after cancelling ambient occlusion. Evaluations are performed on three different temporal spacings.

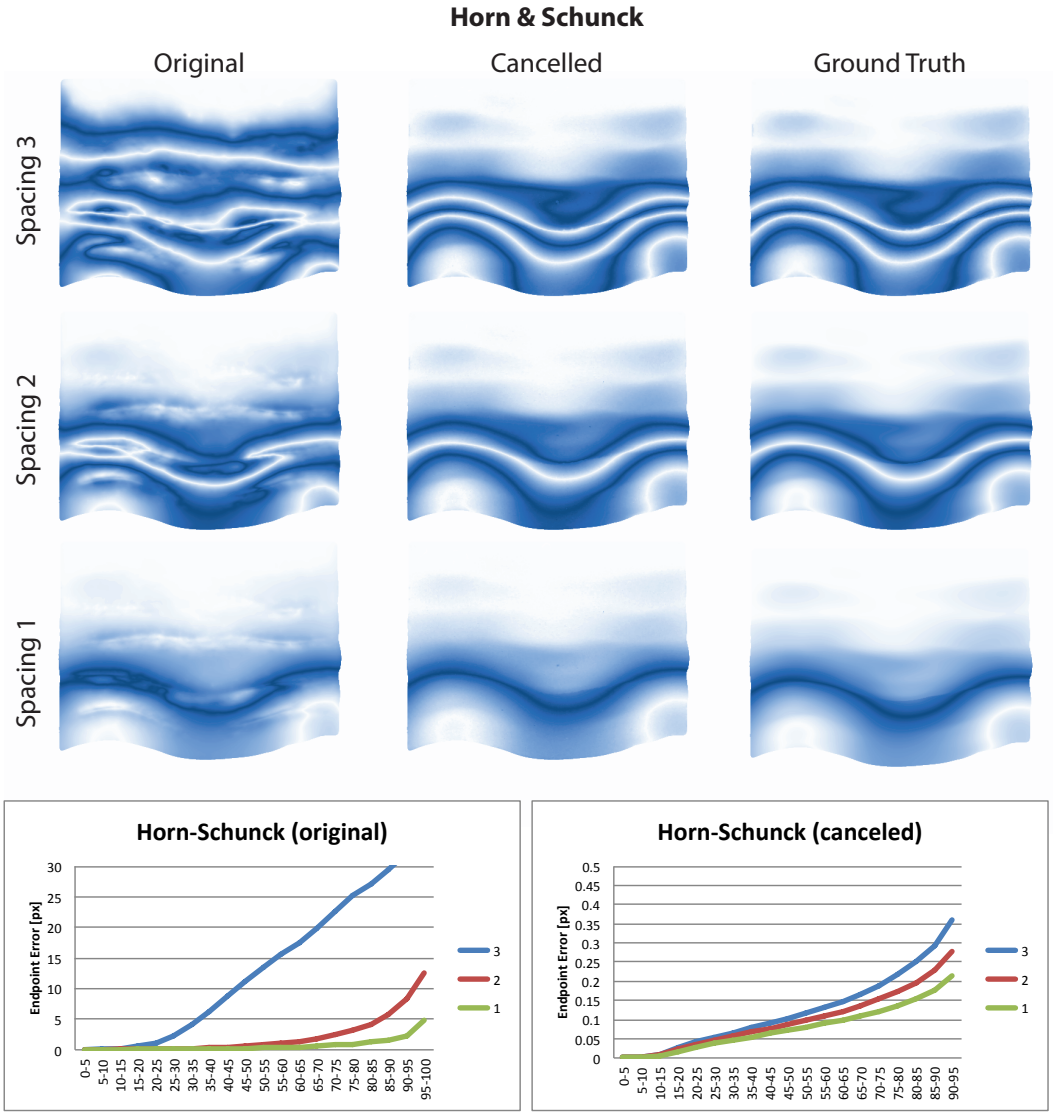


Figure 8.7: *This figure shows an extended evaluation of the improved flow for Horn and Schunck, before and after cancelling ambient occlusion. [TOP]: Flow visualizations for the 3 temporally different spacings shown in Figure 8.13. [BOTTOM]: Percentile plots for the Endpoint Error before and after cancellation for all 3 spacings. Notice the improvement in the upper percentiles, corresponding to the wrinkle region. Also, please note the large scale difference of the plots (60x).*

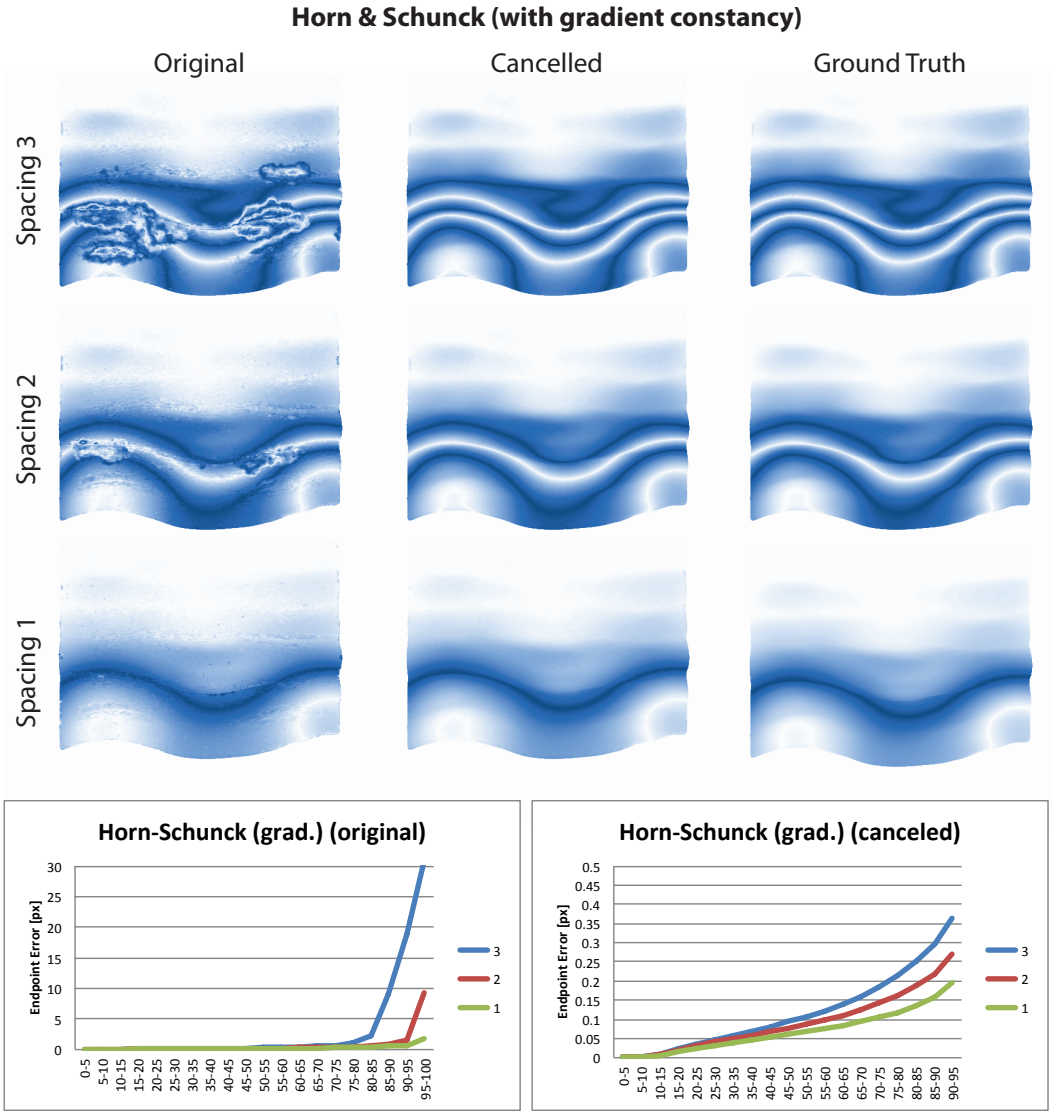


Figure 8.8: *This figure shows an extended evaluation of the improved flow for Horn and Schunck with added gradient constancey, before and after cancelling ambient occlusion. [TOP]: Flow visualizations for the 3 temporally different spacings shown in Figure 8.13. [BOTTOM]: Percentile plots for the Endpoint Error before and after cancellation for all 3 spacings. Notice the improvement in the upper percentiles, corresponding to the wrinkle region. Also, please note the large scale difference of the plots (60x).*

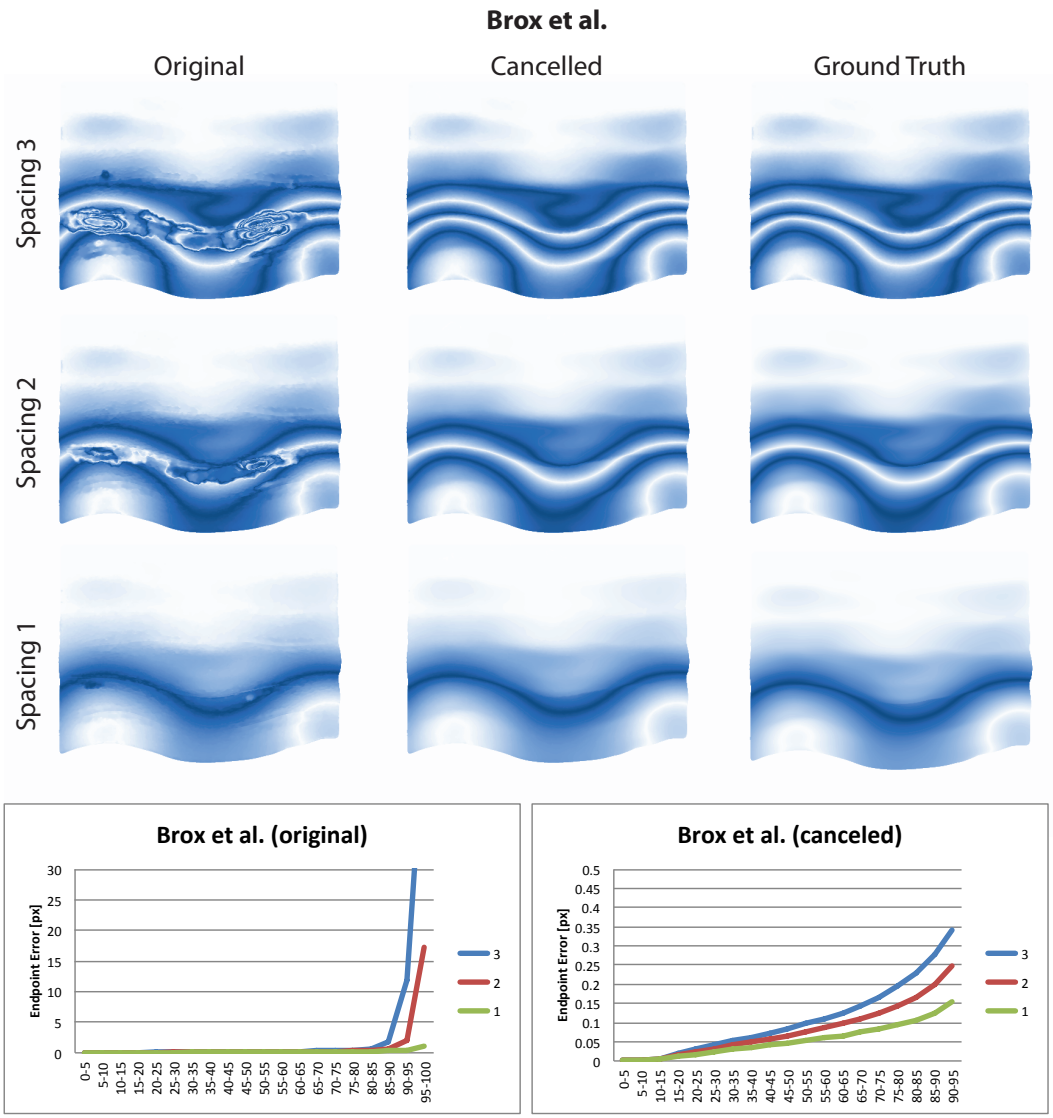


Figure 8.9: *This figure shows an extended evaluation of the improved flow for Brox et al., before and after cancelling ambient occlusion. [TOP]: Flow visualizations for the 3 temporally different spacings shown in Figure 8.13. [BOTTOM]: Percentile plots for the Endpoint Error before and after cancellation for all 3 spacings. Notice the improvement in the upper percentiles, corresponding to the wrinkle region. Also, please note the large scale difference of the plots (60x).*

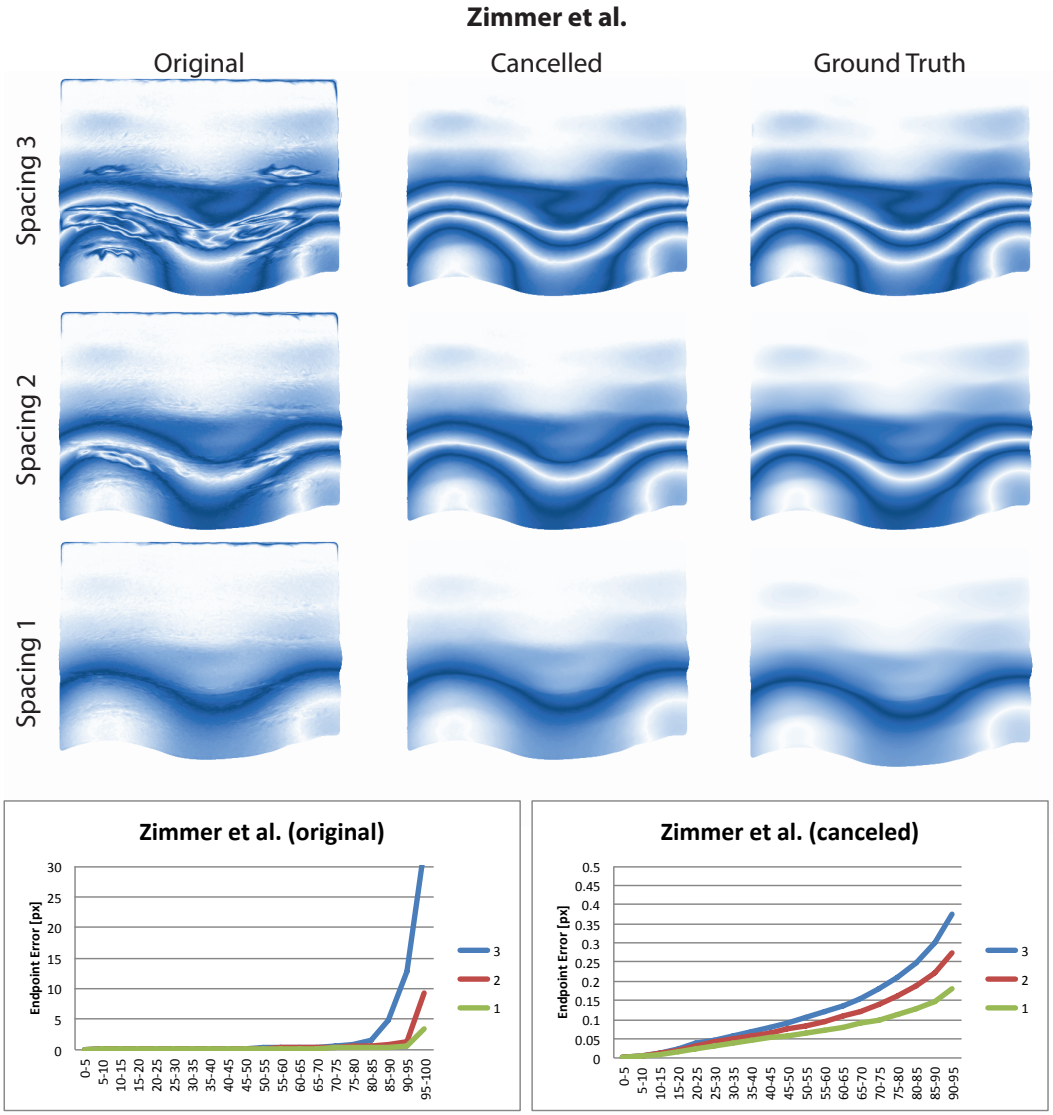


Figure 8.10: *This figure shows an extended evaluation of the improved flow for Zimmer et al., before and after cancelling ambient occlusion. [TOP]: Flow visualizations for the 3 temporally different spacings shown in Figure 8.13. [BOTTOM]: Percentile plots for the Endpoint Error before and after cancellation for all 3 spacings. Notice the improvement in the upper percentiles, corresponding to the wrinkle region. Also, please note the large scale difference of the plots (60x).*

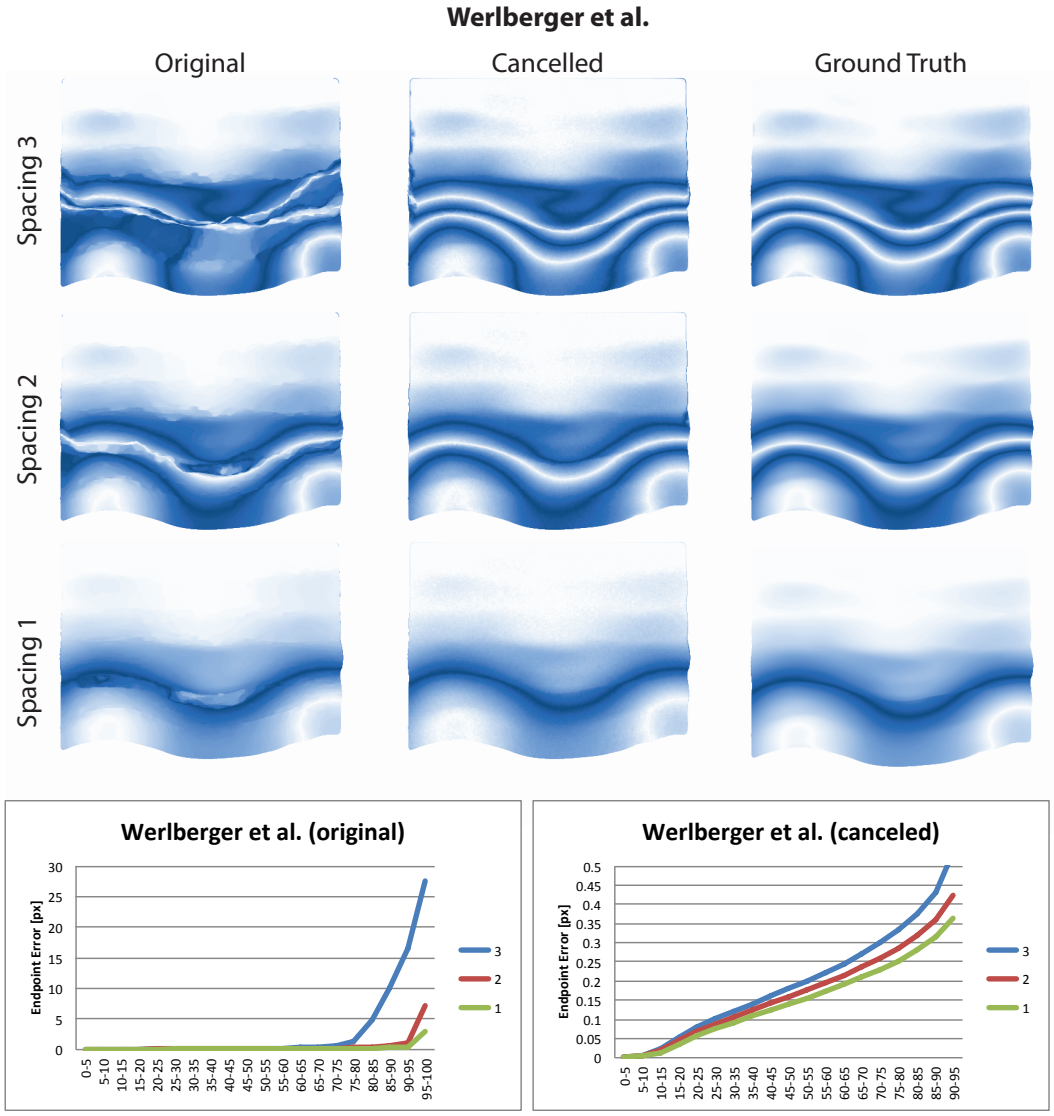


Figure 8.11: *This figure shows an extended evaluation of the improved flow for Werlberger et al., before and after cancelling ambient occlusion. [TOP]: Flow visualizations for the 3 temporally different spacings shown in Figure 8.13. [BOTTOM]: Percentile plots for the Endpoint Error before and after cancellation for all 3 spacings. Notice the improvement in the upper percentiles, corresponding to the wrinkle region. Also, please note the large scale difference of the plots (60x).*

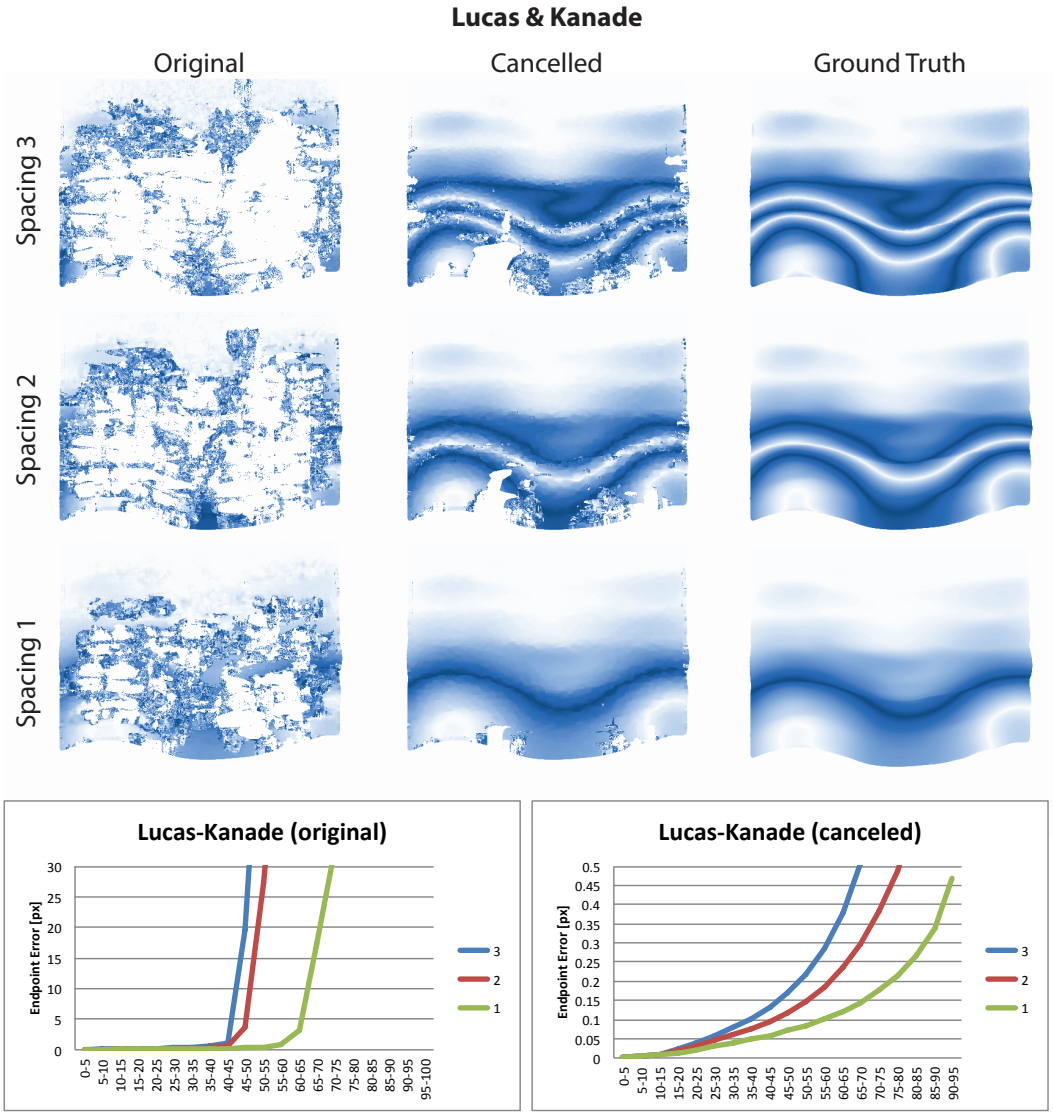


Figure 8.12: *This figure shows an extended evaluation of the improved flow for Lucas and Kanade, before and after cancelling ambient occlusion. [TOP]: Flow visualizations for the 3 temporally different spacings shown in Figure 8.13. [BOTTOM]: Percentile plots for the Endpoint Error before and after cancellation for all 3 spacings. Notice the improvement in the upper percentiles, corresponding to the wrinkle region. Also, please note the large scale difference of the plots (60x).*

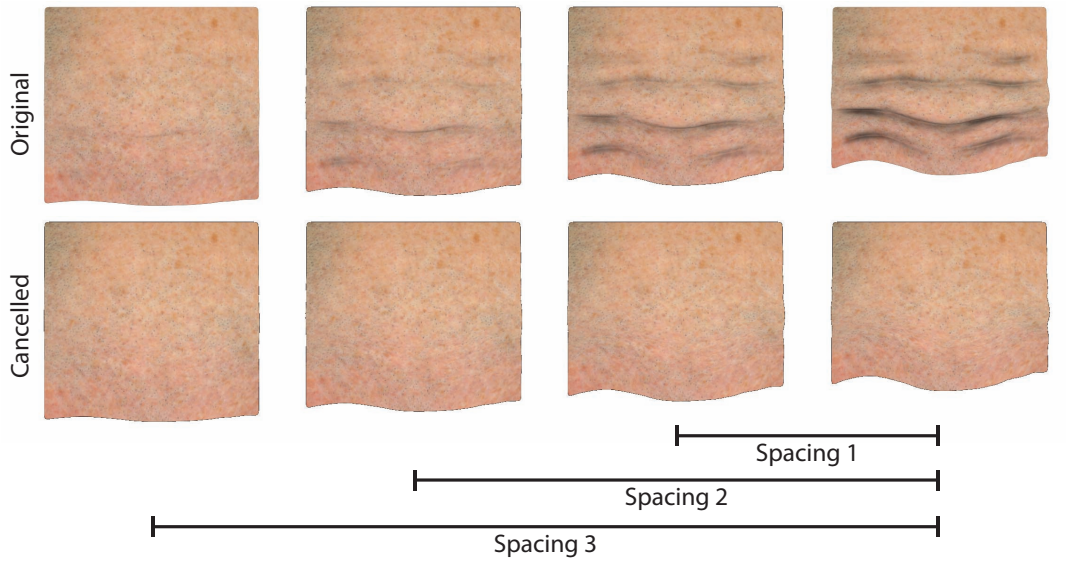


Figure 8.13: Synthetic ground truth - *The ground truth data for flow evaluation is simulated skin wrinkles. The top row shows the original images and the second row is the images after cancelling ambient occlusion. Three temporal spacings are used for the evaluation in this document.*

8.5 Flow Algorithm Parameters

We chose different parameters for original and refined sequences (where the ambient occlusion has been cancelled) to achieve the best results in both scenarios. The parameters we used in our experiments are listed below in Table 8.2.

Here, the parameters have the following meanings:

- α is in general a smoothness weight that balances the contribution of the data term and the smoothness term (regularizer, prior) in the energy function to be minimized. A larger value of α results in a stronger regularization and thus yields a smoother flow field. In the method of Werlberger et al., there is no smoothness weight, but a weight λ is multiplied with the data term. Thus, the corresponding smoothness weight is given by $\alpha = 1/\lambda$. However, the parameter α is also used in the paper of Werlberger et al., but here it determines the shape of an image-driven weighting function that is used in the smoothness term.
- γ determines the weight of the gradient constancy assumption in comparison to the brightness constancy assumption in the data term.

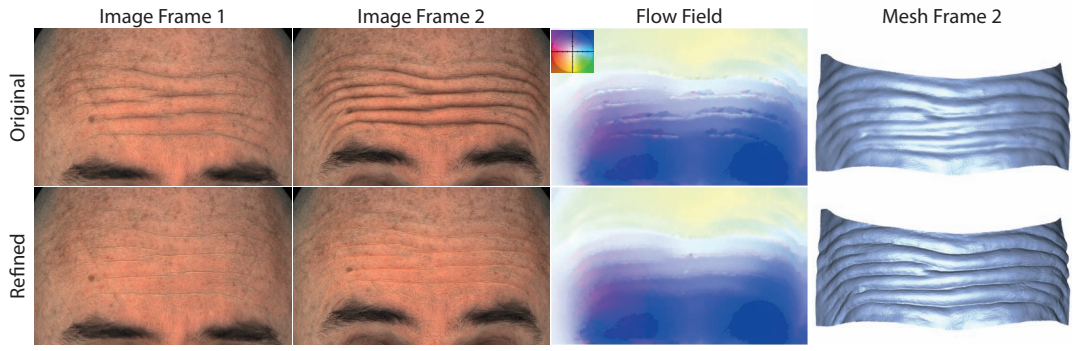


Figure 8.14: Real-World Sequence - *the first row shows the original images along with computed flow field and input shape. The second row shows the cancelled images along with the improved flow field and refined shape after three iterations. The flow fields were computed using the method of Brox et al.*

- β is another parameter used to steer the shape of the image-driven weighting function in the method of Werlberger et al.
- ϵ steers the shape of the robust penaliser function in the smoothness term proposed by Werlberger et al.
- The *window size* parameter in the local Lucas-Kanade method determines the size of a window around each pixel in which the flow field is assumed to be constant. In this sense it serves a similar role as the smoothness weight in the other global methods.
- The *iterations* parameter for the Lucas-Kanade method determines the number of refinement steps performed. Here, the images are warped by the current flow and then a flow correction is computed to iteratively refine the solution.

8.5.1 Real-World Sequence

Figure 8.14 demonstrates the impact of the method on a real-world sequence. Motivated by the analysis of the synthetic data (Table 8.1) we employed the method of Brox et al. to compute the flow as it performed best with similar parameter settings for the original and cancelled images. We ran three overall iterations and the mesh refinement ran for 30 iterations. The runtime of the parallelized C++ implementation was under 3 minutes per frame on a Mac Pro using 8 cores. The size of the images is 539×329 pixels and the mesh

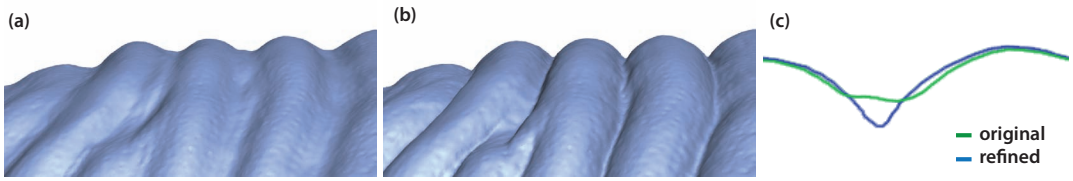


Figure 8.15: Shape Refinement - *(a) and (b) show the original and refined shapes. Figure (c) overlays the silhouette of both shapes for better comparison. Note how the refined shape exhibits the v-shaped valleys and u-shaped ridges characteristic to wrinkles, while the original shape fails to do so.*

consists of roughly 150K vertices. As can be seen, not only the flow field but also the shape of the wrinkles is greatly improved. Figure 8.15 shows that the wrinkles of the original and refined meshes produce substantially different silhouettes.

8.6 Discussion and Conclusion

We have shown that cancelling the ambient occlusion from captured images of a deforming surface can lead to improved motion and surface reconstructions, particularly for high-frequency deformations such as the wrinkles of human skin.

In order to accurately improve reconstructions, the proposed method makes a few assumptions about the captured sequence. First, we assume that the lighting setup is very close to ambient illumination, so that the recorded shading can be closely approximated by ambient occlusion. This is not a severe limitation however, because most capture setups are designed with nearly uniform omni-directional lighting. All capture setups used in this thesis conform to this constraint. In the case of arbitrary non-uniform illumination, we could estimate the environment lighting using a method similar to Wu et al. [Wu et al., 2011a], which we consider future work. The second assumption is that the reconstructed sequence contains at least one frame without any shading, or alternatively the correct shape such that ambient occlusion can be computed and removed. We have referred to this frame as the reference mesh $\mathcal{M}^0$ and reference image I^0. This frame is needed to remove albedo and compute the observed shading in the rest of the sequence (Equation 8.3). For face reconstructions, this can be a neutral pose without wrinkles and typically coincides with the reference frame used by the anchoring algo-

rithm presented in Chapter 7. If no such frame exists, a simple pre-processing step could be applied to search for the brightest occurrence of each vertex in the sequence, assuming that it is un-shaded at some point. We will explore this direction in future work.

This chapter includes a detailed evaluation of the presented technique, using a ground-truth example of a deforming surface patch with known motion. Several well-known optical flow algorithms are shown to benefit from the presented approach. This demonstrates that when considering a specific scenario, like tracking deforming surfaces, modifying the input data in an appropriate way can be a valuable alternative to designing more sophisticated optical flow algorithms that work in that scenario.

We demonstrate the method on a real-world capture sequence of a human face undergoing an expression change. In particular, the proposed refinement process produces much more realistic skin wrinkles. The method can be applied to an existing reconstruction sequence, independently of the reconstruction method or the optical flow algorithm, and it can be easily integrated into new space-time reconstruction algorithms.

CHAPTER 9

Conclusion

This chapter concludes the thesis by summarizing the major contributions, addressing current limitations and suggesting future research directions.

9.1 Contributions

In this thesis we propose a complete system for single-shot, markerless, passive 3D-reconstruction of human faces, including acquisition, calibration and reconstruction.

Chapter 3 discusses acquisition setups. We propose several different setups that are tailored to their respective use cases, such as static or dynamic capture. All methods proposed in this thesis are purely passive in nature, requiring only static diffuse illumination and we describe several illumination setups that provide such conditions, including polarized point lights suited for compact setups, indirect illumination using diffusors for studio appliances, and omnidirectional direct illumination provided by a light stage. A light stage is a spherical construct equipped with light sources that illuminate its center. As part of this thesis, we designed and constructed a versatile multi-purpose light stage, with capabilities extending beyond what is required for

the methods presented in this thesis. The light stage can produce arbitrary illumination conditions that can vary spatially, spectrally and temporally with high dynamic range, making it well suited for other research areas such as appearance capture.

Hardware systems require accurate calibration, which is often a tedious and time consuming process. The goal of the calibration methods proposed in Chapter 4 is to render calibration user friendly and efficient. The proposed camera calibration method uses a simple home-made calibration target and calibrates the setup completely automatically from a single image per camera. The proposed method to geometrically calibrate our light stage uses a mirror sphere and recovers the position and orientation of the light stage without user interaction from a small number of images.

The reconstruction method presented in Chapter 5 is a contribution which brings passive reconstruction of human faces to the same level of quality as active systems. Even though visible in the captured imagery, traditional multi-view stereo (MVS) algorithms are not able to reconstructe fine-scale details, such as small wrinkles and skin pores, because their geometric variation is too small. The key insight presented in this chapter is to filter the input images and extract details at this mesoscopic scale, which are then combined with the coarse geometry produced by the MVS in a unified optimization framework. We call this approach *Mesoscopic Augmentation*. While mesoscopic augmentation is a heuristic method that is not metrically accurate, it greatly increases the visual quality of the results. We performed qualitative and quantitative analysis, demonstrating that the proposed method is on a par with active approaches both in terms of quality and robustness. It is to our knowledge the only passive face capture system to achieve this level of accuracy.

Chapter 6 extends the skin surface reconstruction algorithm presented in Chapter 5 to reconstruct both skin and facial hair in a coupled fashion. Especially in areas where skin is sparsely covered by hair and still visible, it is important to faithfully and correctly reconstruct both skin and hair fibers. In areas of denser hair coverage, skin and individual fibers disappear gradually and the overall look of the hair volume is becoming more important. The method proposed in Chapter 6 implements these observations. It introduces the concept of the *Episurface*, which is accurate where hair is sparse and skin is visible and still plausible where skin is occluded by dense hair coverage. Concurrently, facial hair is reconstructed fiber by fiber where hair coverage is sparse and synthesized in denser areas. It is to our knowledge the first work to capture and reconstruct both skin and hair.

The techniques discussed so far are static. The fact that they require only a single exposure to reconstruct a scan make them an ideal basis for dynamic capture. Chapter 7 extends the skin reconstruction algorithm presented in Chapter 5 to passive, markerless performance capture. The system not only produces the shape of the face at every frame, but also the change of the shape, producing a compatible sequence of meshes. To compute the change, we employ a tailored optical flow algorithm that accurately tracks every point of the geometry in image space. The key insight that makes the technique robust and accurate is the use of *Anchor Frames*, which are based on the observation that certain facial configurations re-occur during a performance. The system identifies these frames and uses them to anchor the computed motion. The concept of anchor frames further allows segmenting a performance into smaller clips and parallelizing reconstruction. It is also a powerful concept to link multiple performances of an actor over an extended period of time.

The shading of a time-varying deformable surface, such as a human face, changes over time as the surface deforms, even under perfect diffuse illumination. These changes are caused by self-shadowing when skin wrinkles or otherwise deforms to block light and pose problems for most optical flow algorithms, such as the one presented in Chapter 7, as they typically expect the intensity of a point to remain constant over time. Chapter 8 proposes a method called *Ambient Occlusion Cancelling* to remove time-varying self-shadowing. The approach complements existing optical flow techniques and we demonstrate improved performance for a variety of well-known optic flow algorithms. In addition, we show how time-varying self-shadowing can be used to improve reconstruction quality.

9.2 Future Work

There are several aspects of facial performance capture that are not addressed in this thesis, leading to many opportunities for future work.

Shape reconstruction We proposed techniques to capture facial skin and sparse facial hair. Different approaches would be required to capture other facial features such as the eyes, teeth or tongue. Especially, modeling and capturing the interface and interplay of eyes, skin and hair (eye-lashes) could be very interesting. Combining the proposed method to reconstruct sparse facial hair with techniques that capture dense hair [Paris et al., 2008] could

extend facial capture to full head acquisition. Extending (sparse) hair capture to the temporal domain would be an equally challenging and interesting topic. Providing anatomically correct episurfaces would further benefit dense hair reconstruction, which could be achieved using strong priors such as morphable models [Blanz and Vetter, 1999]. Finally, extending the proposed high-quality facial capture to full body acquisition would extend its application range. While the technique should work without changes, the larger volume would require a different calibration technique.

Motion capture While the proposed performance capture technique is robust under occlusions, they are not modeled explicitly. Extending the technique to model occlusions would be valuable, especially for the lips. Additionally, handling dis-occlusions by automatically extending the tracked mesh could further be an interesting topic of future research. Areas that are mostly occluded, such as eye-lids or the inside of the mouth, would profit from such a system. One way to cope with missing data would be to introduce regularization in the form of priors. These priors could be computed using physical simulation, which would be a large and very exciting field for future research. The proposed tracking method tracks only within each camera; no information is shared with other cameras, which implies that only areas present in the reference frame can be tracked. Out-of-plane head rotations can therefore not be captured. Extending the algorithm to propagate tracking information between the cameras would be a challenging topic for future research.

Data acquisition So far, all data was acquired with studio-type setups, comprising good cameras and controllable illumination. Extending the methods to work *in the wild* would be a milestone and allow the techniques to be applied, for example, to helmet cameras. For this to be possible, the reconstruction would have to be able to deal with highly distorted images and potentially strong specularities or low light levels. While initial tests on a helmet camera rig were successful, adding additional constraints to the reconstruction method would be beneficial. One possible constraint would be to employ a generic face as prior, or, even better, a prior of the face captured with a studio setup. Furthermore, tracking would have to be improved to be able to handle temporally varying illumination. Combining optical flow with feature based tracking could be an interesting way to approach this problem.

Extension The proposed system generates a vast amount of data. Interesting topics of future work could be how to best handle this data efficiently

and make it available to the user. Data simplification and compression could be interesting topics as well, especially for computer games, where real-time content handling is key. Finally, while reconstructing the geometry of the human face and tracking its motion over time are essential stepping stones toward creating realistic synthetic human faces, additional components such as capturing and representing the correct appearance are necessary as well to reach this goal.

To conclude, this thesis presents a complete pipeline to capture and reconstruct facial performances, including data acquisition and illumination hardware, camera calibration, passive geometry reconstruction of skin and facial hair, as well as dense markerless performance tracking. We believe that the proposed system ranks among the best reconstruction systems to date, both in academia and industry, and hope that the ideas presented in this thesis will inspire future research in the area of time-varying geometry reconstruction.

APPENDIX A

Notation and Glossary

A.1 Operators

$\langle \cdot \rangle$	—	Inner (scalar) product.
$\times$	—	Outer (vector) product.
$*$	—	Convolution.
$\lvert \cdot \rvert$	—	Cardinality for sets, absolute value for scalars.
$\lvert \cdot \rvert_\alpha$	—	Absolute angular difference $[0, \pi[$.
$\lVert \cdot \rVert$	—	Euclidean distance.
$\frac{df}{dx}$	—	Partial derivative of f with respect to x.
$\frac{df}{d\mathbf{x}}$	—	Vector of partial derivatives of f with respect to the components of $\mathbf{x}$.

A.2 Notation

a	—	Scalar.
$\mathbf{a}$	—	Vector.
$\mathbb{R}$	—	The set of real numbers.
$\mathbb{R}^3$	—	Three dimensional space.
$\mathcal{C}$	—	A set of cameras.
$\mathcal{N}(\mu, \sigma)$	—	Gaussian distribution with mean μ and variance σ.
t	—	Time.
a^t	—	Value at time t.
$a^{t \to t+1}$	—	Value from time t to $t+1$.
$\mathbf{p} = (p, q) = (p_x, p_y)$	—	Pixel coordinates ($\mathbb{R}^2$).
$\mathbf{u} = (u, v) = (u_x, u_y)$	—	Differential pixel coordinates ($\mathbb{R}^2$).
$\mathbf{x} = (x, y, z)$	—	Euclidean coordinates in $\mathbb{R}^3$.
$\mathbf{n}$	—	Normal vector in $\mathbb{R}^3$.
$I(\mathbf{p})$	—	Image, defined over pixels $\mathbf{p}$. I will be used for short.
$\mathcal{M}(\mathbf{x})$	—	Mesh defined over vertices $\mathbf{x}$. $\mathcal{M}$ will be used for short.
$\bar{\kappa}$	—	Mean curvature.

A.3 Glossary

CCD	—	Charged-coupled device.
CMOS	—	Complementary metal oxide semiconductor.
CRF	—	Camera response function.
CTF	—	Camera transfer function.
DoG	—	Difference of Gaussians.
DSLR	—	Digital single-lens reflex (camera).
FPS	—	Frames per second.
LED	—	Light-emitting diode.
MVS	—	Multiview stereo.
NCC	—	Normalized cross correlation.
PCA	—	Principal component analysis.
PWM	—	Pulse width modulation.
RMS	—	Root mean squared.
SfS	—	Shape-from-shading.
ToF	—	Time-of-flight.
VNG	—	Variable number of gradients.

A P P E N D I X B

Curriculum Vitae

Personal Information

First Name:	Thabo
Last Name:	Beeler
Address:	Disney Research, Zurich Clausiusstrasse 49 8092 Zurich, Switzerland
Phone:	+41 78 909 85 11
Email:	thabo.beeler@disneyreserach.com
Homepage:	http://graphics.ethz.ch/∼dbeeler/
Nationality:	Swiss
Data and place of birth:	Nov 02 1978 in Roma, Lesotho

Education

since May 2009	joint Ph.D. student at the Swiss Federal Institute of Technology (ETH) and Disney Research Zurich (DRZ).
Oct. 2006 - Feb. 2009	M.Sc. Visual Computing, Swiss Federal Institute of Technology (ETH).
Oct. 2001 - Dec. 2004	Dipl. Ing. Inf. FH, University of Applied Sciences Rapperswil (HSR).
Nov. 2000 - Sept. 2001	Internship, AdNovum Informatik AG, Zurich.
July 1999 - Jan. 2000	Multimedia Producer, SAE College, Zurich.

Employment

May 2009 - Jun. 2012	Research Assistant, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.
WS 2006/2007	Assistant Professor, University of Applied Sciences Rapperswil (HSR), Rapperswil, Switzerland.
Feb. 2005 - Aug. 2006	Research Assistant, University of Applied Sciences Rapperswil (HSR), Rapperswil, Switzerland.
Jan. 2001 - Dec. 2005	CTO, weedbees mediaskills beeler & CO, Frick, Switzerland.
Jun. 1999 - Jan. 2001	Assistant, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Publications

ECCV 2012	T. Beeler, D. Bradley, H. Zimmer, M. Gross: *Improved Reconstruction of Deforming Surfaces by Cancelling Ambient Occlusion*
SIGGRAPH 2012	T. Beeler, B. Bickel, G. Noris, P. Beardsley, S. Marschner, B. Sumner, M. Gross: *Coupled 3D Reconstruction of Sparse Facial Hair and Skin*
SIGGRAPH 2012	B. Bickel, P. Kaufmann, M. Skouras, B. Thomaszewski, D. Bradley, T. Beeler, P. Jackson, S. Marschner, W. Matusik, M. Gross: *Physical Face Cloning*
SIGGRAPH 2011	T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, B. Sumner, M. Gross: *High-quality passive facial performance capture using anchor frames*
ETH Tech-report	T. Beeler, B. Bickel, P. Beardsley, B. Sumner, M. Gross: *High-Quality Single-Shot Capture of Facial Geometry: Implementation Details*
SIGGRAPH 2010	T. Beeler, B. Bickel, P. Beardsley, B. Sumner, M. Gross: *High-Quality Single-Shot Capture of Facial Geometry*
ETH Zurich, master thesis 2009	T. Beeler: *A Gradient Illumination Based Face Scanner*

List of Figures

List of Tables

Bibliography

[Alexander et al., 2009] Alexander, O., Rogers, M., Lambeth, W., Chiang, M., and Debevec, P. E. (2009). The Digital Emily Project: Photoreal Facial Modeling and Animation. *ACM SIGGRAPH 2009 Courses*, pages 12:1–12:15.

[Anuar and Guskov, 2004] Anuar, N. and Guskov, I. (2004). Extracting animated meshes with adaptive motion estimation. In *Vision, Modeling, and Visualization*, pages 63–71.

[Baker et al., 2011] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A Database and Evaluation Methodology for Optical Flow. *IJCV*, 92:1–31.

[Beeler et al., 2010] Beeler, T., Bickel, B., Beardsley, P., Sumner, R. W., and Gross, M. (2010). High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29:40:1–40:9.

[Beeler et al., 2012] Beeler, T., Bickel, B., Noris, G., Marschner, S., Beardsley, P., Sumner, R. W., and Gross, M. (2012). Coupled 3D Reconstruction of Sparse Facial Hair and Skin. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31.

[Beeler et al., 2011] Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R. W., and Gross, M. (2011). High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 30:75:1–75:10.

[Bickel et al., 2007] Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., and Gross, M. (2007). Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3):0730–0301.

[Blais et al., 2004] Blais, F., Picard, M., and Godin, G. (2004). Accurate 3D acquisition of freely moving objects. *3DPVT*, pages 422–429.

[Blake et al., 1985] Blake, A., Zisserman, A., and Knowles, G. (1985). Surface descriptions from stereo and shading. *Image and Vision Computing*, 3(4):183–191.

[Blanz et al., 2003] Blanz, V., Basso, C., Vetter, T., and Poggio, T. (2003). Reanimating Faces in Images and Video. *Computer Graphics Forum (Proc. Eurographics)*, 22(3):641–650.

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

[Bradley et al., 2010] Bradley, D., Heidrich, W., Popa, T., and Sheffer, A. (2010). High Resolution Passive Facial Performance Capture. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29(4):41:1–41:10.

[Bradley et al., 2008] Bradley, D., Popa, T., Sheffer, A., Heidrich, W., and Boubekeur, T. (2008). Markerless Garment Capture. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3):99:1–99:9.

[Brady and Legge, 2009] Brady, M. and Legge, G. E. (2009). Camera calibration for natural image studies and vision research. *J. Opt. Soc. Am. A*, 26(1):30.

[Brox et al., 2004] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36.

[Büttgen et al., 2005] Büttgen, B., Oggier, T., and Lehmann, M. (2005). CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. *Range Imaging Conference*.

[Campbell et al., 2008] Campbell, N. D. F., Vogiatzis, G., Hernández, C., and Cipolla, R. (2008). Using Multiple Hypotheses to Improve Depth-Maps for Multi-View Stereo. In *ECCV*, pages 766–779. ECCV.

[Canny, 1983] Canny, J. (1983). Finding edges and lines in images. Technical report, MIT.

[Canny, 1986] Canny, J. (1986). A Computational Approach to Edge Detection. *PAMI*, PAMI-8(6):679–698.

[Chang and Cheung, 1999] Chang, E. and Cheung, S. (1999). Color filter array recovery using a threshold-based variable number of gradients. *Proc SPIE*, 3650.

[Cheetham et al., 2011] Cheetham, M., Suter, P., and Jäncke, L. (2011). The human likeness dimension of the "uncanny valley hypothesis": Behavioral and functional MRI findings. *Frontiers in Human Neuroscience*, 5.

[Chen et al., 2006] Chen, T., Goesele, M., and Seidel, H.-P. (2006). Mesostructure from Specularity. *CVPR*, 2:1825–1832.

[Dalí and Halsman, 1954] Dalí, S. and Halsman, P. (1954). *Dali's mustache: a photographic interview*. Flammarion.

[De Aguiar et al., 2007] De Aguiar, E., Theobalt, C., Stoll, C., and Seidel, H.-P. (2007). Marker-less Deformable Mesh Tracking for Human Shape and Motion Capture. In *CVPR*, pages 1–8.

[Debevec et al., 2000] Debevec, P. E., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. (2000). Acquiring the Reflectance Field of a Human Face. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 145–156. ACM Press.

[Debevec and Malik, 2008] Debevec, P. E. and Malik, J. (2008). Recovering high dynamic range radiance maps from photographs. *ACM SIGGRAPH 2008 Courses*.

[DeCarlo and Metaxas, 1996] DeCarlo, D. and Metaxas, D. (1996). The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation. In *CVPR*, pages 231–238.

[Delaunoy et al., 2008] Delaunoy, A., Prados, E., Piracés, G., and Pons, J.-P. (2008). Minimizing the multi-view stereo reprojection error for triangular surface meshes. *BMVC*.

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

[Essa et al., 1996] Essa, I., Basu, S., and Darrell, T. (1996). Modeling, tracking and interactive animation of faces and heads using input from video. *Computer Animation*, pages 68–79.

[Forsyth and Ponce, 2002] Forsyth, D. A. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall, 1 edition.

[Fua, 1995] Fua, P. (1995). Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 16(1):35–56.

[Furukawa and Ponce, 2007] Furukawa, Y. and Ponce, J. (2007). Accurate, Dense, and Robust Multi-View Stereopsis. *PAMI*, 32(8):1362–1376.

[Furukawa and Ponce, 2009a] Furukawa, Y. and Ponce, J. (2009a). Accurate camera calibration from multi-view stereo and bundle adjustment. *IJCV*, 84(3):257–268.

[Furukawa and Ponce, 2009b] Furukawa, Y. and Ponce, J. (2009b). Dense 3D Motion Capture for Human Faces. In *CVPR*, pages 1674–1681.

[Fyffe et al., 2011] Fyffe, G., Hawkins, T., Watts, C., Ma, W.-C., and Debevec, P. E. (2011). Comprehensive Facial Performance Capture. *Computer Graphics Forum*, 30(2):425–434.

[Gardner et al., 2003] Gardner, A., Tchou, C., Hawkins, T., and Debevec, P. E. (2003). Linear light source reflectometry. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 22(3):749–758.

[Gargallo et al., 2007] Gargallo, P., Prados, E., and Sturm, P. (2007). Minimizing the reprojection error in surface reconstruction from images. *ICCV*, pages 1–8.

[Gennert and Negahdaripour, 1987] Gennert, M. and Negahdaripour, S. (1987). Relaxing the Brightness Constancy Assumption in Computing Optical Flow. Technical Report A.I. Memo No. 975.

[Ghosh et al., 2011] Ghosh, A., Fyffe, G., Tunwattanapong, B., Busch, J., Yu, X., and Debevec, P. E. (2011). Multiview face capture using polarized spherical gradient illumination. In *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, pages 129:1–129:10. ACM.

[Glencross et al., 2008] Glencross, M., Ward, G., Melendez, F., Jay, C., Liu, J., and Hubbold, R. (2008). A perceptually validated model for surface depth hallucination. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3):59:1–59:8.

[Goesele and Curless, 2006] Goesele, M. and Curless, B. (2006). Multi-view stereo revisited. *CVPR*, 2:2402–2409.

[Goesele et al., 2007] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-View Stereo for Community Photo Collections. *ICCV*, pages 1–8.

[Goldman et al., 2005] Goldman, D., Curless, B., Hertzmann, A., and Seitz, S. (2005). Shape and spatially-varying BRDFs from photometric stereo. In *ICCV*, pages 341–348.

[Grabli et al., 2002] Grabli, S., Sillion, F. X., Marschner, S. R., and Lengyel, J. E. (2002). Image-Based Hair Capture by Inverse Lighting. In *Proceedings of Graphics Interface (GI)*, pages 51–58.

[Guenter et al., 1998] Guenter, B., Grimm, C., Wood, D., Malvar, H., and Pighin, F. (1998). Making Faces. In *Computer Graphics Forum*, pages 55–66.

[Hartley and Zisserman, 2000] Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry*.

[Haussecker and Fleet, 2001] Haussecker, H. W. and Fleet, D. J. (2001). Computing Optical Flow with Physical Models of Brightness Variation. *PAMI*, 23:661–673.

[Hernández and Vogiatzis, 2010] Hernández, C. and Vogiatzis, G. (2010). Self-calibrating a real-time monocular 3D facial capture system. In *3DPVT*.

[Hernández et al., 2008] Hernández, C., Vogiatzis, G., and Cipolla, R. (2008). Shadows in three-source photometric stereo. *ECCV*, 5302:290–303.

[Hiep et al., 2009] Hiep, V. H., Keriven, R., Labatut, P., and Pons, J.-P. (2009). Towards high-resolution large-scale multi-view stereo. *CVPR*, pages 1430–1437.

[Horn, 1970] Horn, B. K. P. (1970). Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. *PhD Thesis MIT*, 21(8):690–706.

[Horn, 1981] Horn, B. K. P. (1981). Determining Optical Flow. *Artificial Intelligence*, 17:185–203.

[Hornung and Kobbelt, 2006] Hornung, A. and Kobbelt, L. (2006). Hierarchical Volumetric Multi-view Stereo Reconstruction of Manifold Surfaces based on Dual Graph Embedding. In *Computer Vision and Pattern Recognition*, pages 503–510. IEEE.

[Iddan, 2001] Iddan, G. (2001). 3D Imaging in the studio (and elsewhere). *SPIE*, 4298.

[Intel, 2012] Intel (2001 (accessed June 26, 2012)). *OpenCV Reference Manual*. http://opencv.willowgarage.com/wiki/.

[Jakob et al., 2009] Jakob, W., Moon, J. T., and Marschner, S. (2009). Capturing hair assemblies fiber by fiber. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 28(3):164:1–164:9.

[Jancosek et al., 2009] Jancosek, M., Shekhovtsov, A., and Pajdla, T. (2009). Scalable multi-view stereo. *ICCV Workshops*, pages 1526–1533.

[Jin et al., 2007] Jin, H., Cremers, D., Wang, D., Prados, E., Yezzi, A., and Soatto, S. (2007). 3-D Reconstruction of Shaded Objects from Multiple Images Under Unknown Illumination. *ICCV*, 76(3):245–256.

[Joly, 2012] Joly, J. (2010 (accessed June 26, 2012)). *Can a polygon make you cry?* http://www.jonathanjoly.com/front.htm.

[Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *SGP*, pages 61–70.

[Kolev et al., 2009] Kolev, K., Klodt, M., Brox, T., and Cremers, D. (2009). Continuous Global Optimization in Multiview 3D Reconstruction. *IJCV*, 84(1):80–96.

[Kraevoy and Sheffer, 2004] Kraevoy, V. and Sheffer, A. (2004). Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23:861–869.

[Lange et al., 1999] Lange, R., Seitz, P., and Biber, A. (1999). Time-of-flight range imaging with a custom solid-state image sensor. In *SPIE*, pages 180–191.

[Leclerc and Bobick, 1991] Leclerc, Y. and Bobick, A. (1991). The direct computation of height from shading. *CVPR*, pages 552–558.

[Lensch et al., 2003] Lensch, H. P. A., Kautz, J., Goesele, M., Heidrich, W., and Seidel, H.-P. (2003). Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.*, 22(2):234–257.

[Levoy et al., 2000] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. (2000). The Digital Michelangelo Project: 3D Scanning of Large Statues. *ACM Trans. Graph. (TOG)*, pages 131–144.

[Li et al., 1993] Li, H., Roivainen, P., and Forcheimer, R. (1993). 3-D Motion Estimation in Model-Based Facial Image Coding. *PAMI*, 15(6):545–555.

[Lin and Ouhyoung, 2005] Lin, I. C. and Ouhyoung, M. (2005). Mirror MoCap: Automatic and efficient capture of dense 3D facial motion parameters from video. *The Visual Computer*, 21(6):355–372.

[Liu et al., 2009] Liu, Y., Cao, X., and Dai, Q. (2009). Continuous depth estimation for multi-view stereo. *CVPR*, pages 2121–2128.

[Lourakis and Argyros, 2009] Lourakis, M. I. A. and Argyros, A. A. (2009). SBA: A software package for generic sparse bundle adjustment. *Transactions on Mathematical Software*, 36(1).

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110.

[Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, pages 674–679.

[Ma et al., 2007] Ma, W.-C., Hawkins, T., Peers, P., Chabert, C.-F., Weiss, M., and Debevec, P. E. (2007). Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. *Rendering Techniques*.

[Ma et al., 2008] Ma, W.-C., Jones, A., Chiang, J.-Y., Hawkins, T., Frederiksen, S., Peers, P., Vukovic, M., Ouhyoung, M., and Debevec, P. E. (2008). Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(5):121:1–121:10.

[Méndez-Feliu and Sbert, 2009] Méndez-Feliu, A. and Sbert, M. (2009). From obscurances to ambient occlusion: A survey. *The Visual Computer*, 25(2):181–196.

[Merrell et al., 2007] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nistér, D., and Pollefeys, M. (2007). Real-Time Visibility-Based Fusion of Depth Maps. *ICCV*, pages 1–8.

[Meyer et al., 2003] Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2003). *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. Springer-Verlag.

[Molnar et al., 2010] Molnar, J., Chetverikov, D., and Fazekas, S. (2010). Illumination-robust variational optical flow using cross-correlation. *CVIU*, 114:1104–1114.

[Mori, 1970] Mori, M. (1970). Bukimi no tani — The uncanny valley. *Energy*, pages 7:33–35.

[Nehab et al., 2005] Nehab, D., Rusinkiewicz, S., Davis, J., and Ramamoorthi, R. (2005). Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3):536–543.

[Paris et al., 2004] Paris, S., Briceño, H. M., and Sillion, F. X. (2004). Capture of hair geometry from multiple images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23:712–719.

[Paris et al., 2008] Paris, S., Chang, W., Kozhushnyan, O. I., Jarosz, W., Matusik, W., Zwicker, M., and Durand, F. (2008). Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27:30:1–30:9.

[Pighin et al., 1999] Pighin, F. H., Szeliski, R., and Salesin, D. (1999). Resynthesizing Facial Animation through 3D Model-based Tracking. In *ICCV*, pages 143–150.

[Pons et al., 2005] Pons, J.-P., Keriven, R., and Faugeras, O. (2005). Modelling Dynamic Scenes by Registering Multi-View Image Sequences. *CVPR*, 2:822–827.

[Popa et al., 2010] Popa, T., South-Dickinson, I., Bradley, D., Sheffer, A., and Heidrich, W. (2010). Globally Consistent Space-Time Reconstruction. *Computer Graphics Forum*, 29(5):1633–1642.

[Posdamer and Altschuler, 1982] Posdamer, J. and Altschuler, D. (1982). Surface measurement by space-encoded projected beam systems. *Computer graphics and image processing*, 18(1):1–17.

[Proesmans et al., 1996] Proesmans, M., Van Gool, L., and Oosterlinck, A. (1996). One-shot active 3D shape acquisition. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, pages 336–340 vol.3.

[Rav-Acha et al., 2008] Rav-Acha, A., Kohli, P., Rother, C., and Fitzgibbon, A. W. (2008). Unwrap Mosaics: A new representation for video editing. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3):17:1–17:11.

[Robert and Deriche, 1996] Robert, L. and Deriche, R. (1996). Dense Depth Map Reconstruction: A Minimization and Regularization Approach which Preserves Discontinuities. In *ECCV*, pages 439–451.

[Saint-Marc et al., 1991] Saint-Marc, P., Jezouin, J., and Medioni, G. (1991). A versatile PC-based range finding system. *IEEE Transactions on Robotics and Automation*, 7(2):250–256.

[Salvi et al., 2004] Salvi, J., Pagès, J., and Batlle, J. (2004). Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849.

[Samaras et al., 2000] Samaras, D., Metaxas, D., Fua, P., and Leclerc, Y. (2000). Variable albedo surface reconstruction from stereo and shape from shading. *CVPR*, 1:480–487.

[Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *IJCV*, 47:7–42.

[Scharstein and Szeliski, 2003] Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *CVPR*, pages I–195–I–202. IEEE Comput. Soc.

[Seitz and Baker, 2009] Seitz, S. M. and Baker, S. (2009). Filter Flow. In *ICCV*, pages 143–150.

[Seitz et al., 2006] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. *CVPR*, 1:519–528.

[Sharf et al., 2008] Sharf, A., Alcantara, D. A., Lewiner, T., Greif, C., Sheffer, A., Amenta, N., and Cohen-Or, D. (2008). Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(5):110:1–110:10.

[Son and Davis, 2006] Son, T. and Davis, L. (2006). 3D Surface Reconstruction Using Graph Cuts with Surface Constraints. *ECCV*, 3952:219–231.

[Sormann et al., 2007] Sormann, M., Zach, C., Bauer, J., Karner, K., and Bishof, H. (2007). Watertight multi-view reconstruction based on volumetric graph-cuts. *Image Analysis*, 4522:393–402.

[Strecha et al., 2006] Strecha, C., Fransens, R., and Gool, L. V. (2006). Combined depth and outlier estimation in multi-view stereo. *CVPR*, 2:2394–2401.

[Sumner and Popović, 2004] Sumner, R. W. and Popović, J. (2004). Deformation transfer for triangle meshes. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23:399–405.

[Svoboda et al., 2005] Svoboda, T., Martinec, D., and Pajdla, T. (2005). A Convenient Multicamera Self-Calibration for Virtual Environments. *Presence: Teleoper. Virtual Environ.*, 14(4):407–422.

[Szeliski, 1999] Szeliski, R. (1999). A multi-view approach to motion and stereo. *CVPR*, 1.

[TexasInstruments, 2012] TexasInstruments (2008 (accessed June 26, 2012)). *1.6MHz, 1A Constant Current Buck LED Driver with Internal Compensation*. http://www.ti.com/product/lm3405a.

[Torralba and Freeman, 2003] Torralba, A. and Freeman, W. T. (2003). Properties and Applications of Shape Recipes. *CVPR*, 2:383–390.

[Tsai, 1987] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344.

[Vlasic et al., 2005] Vlasic, D., Brand, M., Pfister, H., and Popović, J. (2005). Face transfer with multilinear models. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3).

[Vogiatzis et al., 2007] Vogiatzis, G., Hernández, C., Torr, P. H. S., and Cipolla, R. (2007). Multiview Stereo via Volumetric Graph-Cuts and Occlusion Robust Photo-Consistency. *PAMI*, 29(12):2241–2246.

[Wand et al., 2009] Wand, M., Adams, B., Ovsjanikov, M., Berner, A., Bokeloh, M., Jenke, P., Guibas, L., Seidel, H.-P., and Schilling, A. (2009). Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data. *ACM Trans. Graph.*, 28(2):15:1–15:15.

[Wang et al., 2004] Wang, Y., Huang, X., Lee, C.-S., Zhang, S., Li, Z., Samaras, D., Metaxas, D., Elgammal, A., and Huang, P. S. (2004). High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. *Comp. Graphics Forum*, 23(3):677–686.

[Wedel et al., 2008] Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2008). An improved algorithm for TV-L1 optical flow. *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45.

[Wei et al., 2005a] Wei, Y., Ofek, E., and Quan, L. (2005a). Modeling hair from multiple views. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3):816–820.

[Wei et al., 2005b] Wei, Y., Ofek, E., Quan, L., and Shum, H.-Y. (2005b). Modeling hair from multiple views. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3):816–820.

[Weise et al., 2007] Weise, T., Leibe, B., and Gool, L. V. (2007). Fast 3D Scanning with Automatic Motion Compensation. *CVPR*, pages 1–8.

[Werlberger et al., 2010] Werlberger, M., Pock, T., and Bischof, H. (2010). Motion Estimation with Non-Local Total Variation Regularization. In *CVPR*, pages 2464–2471.

[Weyrich et al., 2006] Weyrich, T., Matusik, W., Pfister, H., Bickel, B., Donner, C., Tu, C., McAndless, J., Lee, J., Ngan, A., Jensen, H. W., and Gross, M. (2006). Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3):1013–1024.

[Williams, 1990] Williams, L. (1990). Performance driven facial animation. *Computer Graphics Forum*, 24(3):235–242.

[Wilson et al., 2010] Wilson, C., Ghosh, A., Peers, P., Chiang, J.-Y., Busch, J., and Debevec, P. E. (2010). Temporal upsampling of performance geometry using photometric alignment. *ACM Trans. Graph. (TOG)*, 29(2):17:1–17:11.

[Winkler et al., 2008] Winkler, T., Hormann, K., and Gotsman, C. (2008). Mesh massage. *The Visual Computer*, 24:775–785.

[Woodford et al., 2008] Woodford, O. J., Torr, P. H. S., Reid, I. D., and Fitzgibbon, A. W. (2008). Global stereo reconstruction under second order smoothness priors. *CVPR*, pages 1–8.

[Woodham, 1980] Woodham, R. J. (1980). Photometric method for determining surface orientation. *Optical engineering*.

[Wu et al., 2011a] Wu, C., Varanasi, K., Liu, Y., Seidel, H.-P., and Theobalt, C. (2011a). Shading-based Dynamic Shape Refinement from Multi-view Video under General Illumination. In *ICCV*, pages 1108–1115.

[Wu et al., 2011b] Wu, C., Wilburn, B., Matsushita, Y., and Theobalt, C. (2011b). High-quality shape from multi-view stereo and shading under general illumination. *CVPR*, pages 969–976.

[Zach et al., 2007] Zach, C., Pock, T., and Bischof, H. (2007). A globally optimal algorithm for robust TV-L1 range image integration. *ICCV*, pages 1–8.

[Zaharescu et al., 2007] Zaharescu, A., Boyer, E., and Horaud, R. (2007). TransforMesh: a topology-adaptive mesh-based approach to surface evolution. *ACCV*, 2:166–175.

[Zhang et al., 2002] Zhang, L., Curless, B., and Seitz, S. M. (2002). Rapid shape acquisition using color structured light and multi-pass dynamic programming. *3DPVT*, pages 24–36.

[Zhang et al., 2003] Zhang, L., Curless, B., and Seitz, S. M. (2003). Spacetime stereo: shape recovery for dynamic scenes. *CVPR*, 2:367–374.

[Zhang et al., 2004] Zhang, L., Snavely, N., Curless, B., and Seitz, S. (2004). Spacetime Faces: High Resolution Capture for Modeling and Animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):548–558.

[Zhang et al., 1999] Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M. (1999). Shape-from-shading: a survey. *PAMI*, 21(8):690–706.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *PAMI*, 22(11):1330–1334.

[Zhukov et al., 1998] Zhukov, S., Iones, A., and Kronin, G. (1998). An ambient light illumination model. *Proc. of Eurographics Workshop on Rendering*, pages 45–55.

[Zickler et al., 2005] Zickler, T., Ramamoorthi, R., Enrique, S., and Belhumeur, P. N. (2005). Reflectance sharing: predicting appearance from a sparse set of images of a known shape. *PAMI*, 28(8):1287–1302.

[Zimmer et al., 2011] Zimmer, H., Bruhn, A., and Weickert, J. (2011). Optic Flow in Harmony. *IJCV*, 93(3):368–388.

[Zwicker et al., 2002] Zwicker, M., Pfister, H., van Baar, J., and Gross, M. (2002). EWA splatting. *IEEE Trans. Visualization and Computer Graphics*, 8(3):223–238.