Inverse Rendering for Scene Reconstruction in General Environments

Chenglei Wu

Saarbrücken, Germany

Dissertation
zur Erlangung des Grades des
Doktors der Ingenieurswissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

March 2014

Dekan - Dean:

Prof. Dr. Markus Bläser
Saarland University
Saarbrücken, Germany

Kolloquiums - Defense

Datum - Date
July 10, 2014, in Saarbrücken

Vorsitzender - Head of Colloquium:
Prof. Dr. Bernt Schiele

Prüfer - Examiners:

Prof. Dr. Christian Theobalt

Prof. Dr. Hans-Peter Seidel

Prof. Dr. Markus Gross

Protokoll - Reporter:

Dr. Janick Martinez Esturo

To my loving son, Chongxi.

Abstract

Demand for high-quality 3D content has been exploding recently, owing to the advances in 3D displays and 3D printing. However, due to insufficient 3D content, the potential of 3D display and printing technology has not been realized to its full extent. Techniques for capturing the real world, which are able to generate 3D models from captured images or videos, are a hot research topic in computer graphics and computer vision. Despite significant progress, many methods are still highly constrained and require lots of prerequisites to succeed. Marker-less performance capture is one such dynamic scene reconstruction technique that is still confined to studio environments. The requirements involved, such as the need for a multi-view camera setup, specially engineered lighting or green-screen backgrounds, prevent these methods from being widely used by the film industry or even by ordinary consumers.

In the area of scene reconstruction from images or videos, this thesis proposes new techniques that succeed in general environments, even using as few as two cameras. Contributions are made in terms of reducing the constraints of marker-less performance capture on lighting, background and the required number of cameras. The primary theoretical contribution lies in the investigation of light transport mechanisms for high-quality 3D reconstruction in general environments. Several steps are taken to approach the goal of scene reconstruction in general environments. At first, the concept of employing inverse rendering for scene reconstruction is demonstrated on static scenes, where a high-quality multi-view 3D reconstruction method under general unknown illumination is developed. Then, this concept is extended to dynamic scene reconstruction from multi-view video, where detailed 3D models of dynamic scenes can be captured under general and even varying lighting, and in front of a general scene background without a green screen. Finally, efforts are made to reduce the number of cameras employed. New performance capture methods using as few as two cameras are proposed to capture high-quality 3D geometry in general environments, even outdoors.

Kurzfassung

Die Nachfrage nach qualitativ hochwertigen 3D Modellen ist in letzter Zeit, bedingt durch den technologischen Fortschritt bei 3D-Wiedergabegeräten und -Druckern, stark angestiegen. Allerdings konnten diese Technologien wegen mangelnder Inhalte nicht ihr volles Potential entwickeln. Methoden zur Erfassung der realen Welt, welche 3D-Modelle aus Bildern oder Videos generieren, sind daher ein brandaktuelles Forschungsthema im Bereich Computergrafik und Bildverstehen. Trotz erheblichen Fortschritts in dieser Richtung sind viele Methoden noch stark eingeschränkt und benötigen viele Voraussetzungen um erfolgreich zu sein. Markerloses Performance Capturing ist ein solches Verfahren, das dynamische Szenen rekonstruiert, aber noch auf Studio-Umgebungen beschränkt ist. Die spezifischen Anforderung solcher Verfahren, wie zum Beispiel einen Mehrkameraaufbau, maßgeschneiderte, kontrollierte Beleuchtung oder Greenscreen-Hintergründe verhindern die Verbreitung dieser Verfahren in der Filmindustrie und besonders bei Endbenutzern.

Im Bereich der Szenenrekonstruktion aus Bildern oder Videos schlägt diese Dissertation neue Methoden vor, welche in beliebigen Umgebungen und auch mit nur wenigen (zwei) Kameras funktionieren. Dazu werden Schritte unternommen, um die Einschränkungen bisheriger Verfahren des markerlosen Performance Capturings im Hinblick auf Beleuchtung, Hintergründe und die erforderliche Anzahl von Kameras zu verringern. Der wichtigste theoretische Beitrag liegt in der Untersuchung von Licht-Transportmechanismen für hochwertige 3D-Rekonstruktionen in beliebigen Umgebungen. Dabei werden mehrere Schritte unternommen, um das Ziel der Szenenrekonstruktion in beliebigen Umgebungen anzugehen. Zunächst wird die Anwendung von inversem Rendering auf die Rekonstruktion von statischen Szenen dargelegt, indem ein hochwertiges 3D-Rekonstruktionsverfahren aus Mehransichtsaufnahmen unter beliebiger, unbekannter Beleuchtung entwickelt wird. Dann wird dieses Konzept auf die dynamische Szenenrekonstruktion basierend auf Mehransichtsvideos erweitert, wobei detaillierte 3D-Modelle von dynamischen Szenen unter beliebiger und auch veränderlicher Beleuchtung vor einem allgemeinen Hintergrund

ohne Greenscreen erfasst werden. Schließlich werden Anstrengungen unternommen die Anzahl der eingesetzten Kameras zu reduzieren. Dazu werden neue Verfahren des Performance Capturings, unter Verwendung von lediglich zwei Kameras vorgeschlagen, um hochwertige 3D-Geometrie im beliebigen Umgebungen, sowie im Freien, zu erfassen.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Christian Theobalt, for introducing me to the topic of performance capture, for his guidance and support throughout my PhD, and for helping me to build my research skills. His guidance helped me in all the time of research and writing of this thesis. Without him, this thesis would not be possible. I am proud of being part of his group, the graphics, vision and video group, and will always remember him as a great mentor with deep knowledge.

I would also like to thank Prof. Dr. Hans-Peter Seidel for creating such a truly remarkable research environment in the computer graphics group at MPI. It is really a honor for me to work in such an outstanding group.

I am also thankful to Dr. Levi Valgaerts, who was my second mentor. His endless patience in answering my technical questions, his academic rigour with our research and his help in writing and revising manuscripts, make my PhD much less difficult.

Furthermore, I would like to thank Prof. Dr. Markus Gross who kindly agreed to serve as an external reviewer, which I am grateful for.

I also owe special gratitude to other research collaborators: Dr. Yasuyuki Matsushita and Dr. Bennett Wilburn, who introduced me to the field of shading-related techniques and helped me make my first steps as a researcher in this field; Dr. Kiran Varanasi and Dr. Yebin Liu for their guidance and unconditional commitment in the projects; Dr. Carsten Stoll for his profound knowledge of performance capture; Prof. Dr. Andres Bruhn, Guannan Li, and Pablo Garrido.

I highly appreciate the time that Dr. Levi Valgaerts, Dr. James Tompkin, Dr. Kwang In Kim, Dr. Carsten Stoll and Dr. Kiran Varanasi spent on proofreading parts of this thesis. Special thanks to Margaret De Lap for her help in proofreading the thesis. Also thanks to Dr. Christian Richardt for proofreading parts of the thesis and his help in translating the abstract.

I would also like to express my sincere thanks to the administrative staff members, Sabine Budde and Ellen Fries from MPI, and Hanna Loger and Diane Chlupka from Intel VCI. They are always kind and generous in supporting me with their professional work; heartfelt thanks for their excellent work. Many thanks to my officemate Ahmed Elhayek. It was great fun to share the office with him, and he deserves a medal for coping with me and my cluttered boxes.

Furthermore, I owe thanks to all my colleagues in the computer graphics group at MPI. It is these colleagues who make MPI such a wonderful place to pursue my research. I cannot name all of them, but I would like to especially thank the following people: Nils Hasler, Andreas Baak, Thomas Helten, Helge Rhodin, Miguel Granados, Martin Grochulla, and Michal Richter.

Finally, I would like to thank my parents, Jianchang Wu and Xijuan Ren, for their unremitting support. They have always stood by me and encouraged me throughout my whole life. Above all, I would like to thank my wife, Bing, who supports me in all that I do. Her unwavering love is the source of my original inspiration.

Contents

Chapter 1

Introduction

In the last decades, computer generated content has become very popular in the entertainment industry, e.g. films or video games. Especially for some dynamic content, like virtual characters, these scenes are particularly important, but difficult to model. Traditionally, to create such content, an artist would need to create the model manually, and then hand-craft the animation, the high-quality surface detail and even the surface material properties, which are painstakingly time-consuming processes. New techniques to improve both the quality of the content as well as the ease of creation are in strong demand from the industry. Therefore, the technology to create such content has been a hot research area in computer graphics and computer vision for many years. Real world capture from images or videos is one of the most important techniques able to create realistic models for both static and dynamic scenes.

As one of the real-world capture methods for dynamic scenes, performance capture has achieved great success in recent years, which can be generally distinguished into marker-based methods and marker-less methods. Marker-based methods use actively placed fiducial markers to track the 3D positions of these sparse scene points in order to estimate the coarse skeletal motion or a coarse 3D model. The requirement to use markers severely limits the range of use for these methods. In comparison, marker-less performance capture methods are able to capture much richer and far more expressive models from multiple video recordings Bradley *et al.* (2010); de Aguiar *et al.* (2008); Gall *et al.* (2009); Vlasic *et al.* (2008), since they are able to reconstruct detailed motion, dense dynamic geometry and even rich surface appearance. However, these methods have not yet found their way into many practical feature film productions. One of the major reasons is that most existing methods are still limited to work in a studio

environment, with controlled lighting, controlled background, and an expensive and complicated multi-view camera setup, which makes these methods difficult to deploy on set. The ability to capture detailed 3D models of dynamic scenes in a natural and general environment, e.g. on the movie production set, rather than in a separate stage in the studio, would have a variety of important benefits and would pave the way for many relevant applications of marker-less performance capture. Moreover, a performance capture method which works with just a lightweight setup, i.e. using as few cameras as possible, would further make the technique applicable not only for professional movie producers, but also as a tool which can be generally employed by average consumers or home users to capture myriad 3D content from their daily lives.

In this thesis, we propose new techniques in the area of scene reconstruction from images or videos, especially new techniques in marker-less performance capture, that are able to capture high-quality 3D geometry without the requirements for controlled lighting or controlled background, and that succeed even using a very sparse camera setup. Insights are gained from inverse rendering, which tries to infer lighting, geometry or reflectance from captured images. The main technical contribution of this thesis is to propose new algorithmic solutions for inverse rendering at previously unseen complexity in general environments, and advance techniques in 3D shape reconstruction, high-quality dynamic detail estimation and skeletal motion tracking. An overview of these techniques is given next.

1.1 Overview

This thesis proposes new scene reconstruction methods which succeed in less constrained or even general environments. By investigating mechanisms for light transport in general environments, we are aiming to make high-quality performance capture succeed for general scenes. We approach this goal in several steps. We first prove that the concept of inverse rendering works for the reconstruction of static scenes from multi-view input, where a high-quality shape reconstruction method that succeeds under general unknown illumination is developed. We then extend this concept to dynamic scene reconstruction, still indoors and using eight or more cameras but under fewer constraints. Finally, we push towards reducing the number of cameras required, using as few as two cameras for motion and shape reconstruction in general environments, even outdoors.

1.1.1 Static 3D Reconstruction from Multi-view Images under General Illumination

In part I, we investigate the concept of inverse rendering for scene reconstruction on static scenes, and propose a method for high-quality geometry reconstruction from multi-view images by combining multi-view stereo and shape-from-shading (SfS) under general and uncontrolled illumination. As is known from previous literature, multi-view stereo reconstructs 3D geometry well for sufficiently textured scenes, but often fails to recover high-frequency surface detail, particularly for smoothly shaded surfaces. Alternatively, shape-from-shading methods can recover fine detail from shading variations. However, most shading-based estimation methods only succeed under very restricted or controlled illumination, and it is also non-trivial to apply SfS alone to multi-view data. In this part, by assuming Lambertian surface reflectance with uniform albedo, inverse rendering is exploited to develop a new method, which combines the stereo cue and shading cue in an appropriate way, for high-quality 3D reconstructions under general and uncontrolled illumination. The high quality results generated by this method demonstrate the validity of our concept.

1.1.2 Performance Capture from Multi-view Video under General Illumination

In part II, we extend the use of inverse rendering to dynamic scene reconstruction, specifically to full-body performance capture, using a multi-view camera setup. Two steps are taken to reconstruct detailed models of dynamic scenes in a general environment. At first, in Chapter 5, we exploit the inverse rendering for high-frequency geometric detail estimation in a spatio-temporally coherent way for Lambertian surfaces with spatially varying albedos. Previous performance capture methods de Aguiar *et al.* (2008); Vlasic *et al.* (2008) show plausible deformations up to medium scale detail, but often lack true detail at the finest level. In these methods, a static laser scan is usually deformed to mimic the motion of the real scene, but any fine scale detail thus obtained appears baked into the surface in the rest of the frames and does not capture the true surface detail, e.g. soft wrinkles on clothes. In comparison, our method takes a step forward by capturing the true fine-scale dynamic detail. Besides, the ability to work under general and uncontrolled illumination also substantially relaxes the constraint on specially engineered lighting, e.g. a light stage Vlasic *et al.* (2009),

for high-quality performance capture. However, our method still employs an off-the-shelf performance capture method Gall *et al.* (2009) for low-frequency geometry reconstruction, which is constrained by the need for constant lighting and a green-screen background.

Thus, in Chapter 6, we present a new performance capture method to work wholly under general and varying illumination, and using a general background without a green screen. This is achieved by analyzing shading information for skeletal motion tracking and low-frequency geometry reconstruction, as well as high-frequency geometry estimation. The main technical contribution is that, by an analysis-through-synthesis framework, differential 3D human pose changes from the previous time step can be expressed in terms of constraints on the visible image displacements derived from shading cues, surface albedos and scene illumination. By assuming the Lambertian model of reflectance, the incident illumination at each frame is estimated jointly with pose parameters, enabling the method to work under varying lighting, where the previous methods Gall *et al.* (2009) would fail. In addition, the proposed method is independent of image silhouettes, and is thus applicable in cases where background segmentation cannot be easily performed. By combining it with a dynamic shape refinement step, a new high-quality performance capture method is developed to work in a general environment, even though a multi-view camera setup is still needed.

1.1.3 Binocular Performance Capture

Our new ability to estimate lighting, shape and motion from video in general environments enables us to improve many elementary algorithmic aspects of performance capture. In part III, we show how these algorithms help us to drastically reduce the number of input cameras needed, while still being able to reconstruct detailed 3D models in general unconstrained scenes, even outdoors.

In Chapter 7, a new binocular facial performance capture method is featured. In this method, the dynamic 3D geometry of the facial performance is firstly reconstructed on a coarse level by tracking the surface of a face template based on scene-flow constraints. Then, an improved shape refinement algorithm, which is tailored specifically for face capture, is introduced to obtain the fine-scale detail. The proposed method can capture high-quality geometry of expressive facial performances in an uncontrolled environment, even from a hand-held consumer stereo camera under changing illumination outdoors.

In Chapter 7, efforts are made to reduce the number of cameras needed to capture full body performances in a general environment. In detail, we propose a new full-body performance capture method that is able to track skeletal motion and detailed surface geometry of one or more actors from footage recorded with a stereo rig which is allowed to move. This method succeeds in general sets with uncontrolled background and uncontrolled illumination. In this method, we also generalize the Lambertian reflectance assumption to general surface reflectance, which also models the non-Lambertian reflectance, to estimate the skeletal motion and to refine the fine scale surface geometry. We also develop a new foreground segmentation approach that combines appearance, stereo and pose tracking results to segment out actors from the background. Appearance, segmentation and motion cues are combined in a new pose optimization framework that is robust under uncontrolled lighting, uncontrolled background and very sparse camera views. This is the first method able to achieve high-quality performance capture under such unconstrained conditions, which approach typical movie production sets.

1.1.4 Other Applications

In part IV, we introduce two applications which demonstrate the techniques proposed in previous chapters. One is relightable performance capture, which also captures the surface reflectance in addition to the dynamic geometry. The other is to capture dynamic face geometry from only monocular video. As these two applications contain techniques beyond the scope of this thesis, we will only focus on the parts related to the thesis.

1.2 Contributions

The performance capture methods presented in this thesis have been presented at international conferences and published in international journals Garrido *et al.* (2013); Li *et al.* (2013); Valgaerts *et al.* (2012b); Wu *et al.* (2011a,b, 2012, 2013). This thesis presents an extended version of these methods (Chapters 4- 9). To sum up, the key contributions are:

- A new shape reconstruction method that combines multi-view stereo and shape-from-shading under general and uncalibrated illumination to achieve very high-quality 3D reconstructions, which is much better than the stereo

based approaches and rivals laser range scans (Chapter 4). Specifically, a new multi-view shading constraint is presented. An adaptive anisotropic smoothness term for preserving high-frequency details while filtering out noise is proposed. In addition, an adaptive computation approach is developed to take the complexity of lighting and visibility estimates into account at each surface point to achieve a good compromise between efficiency and accuracy. This work has been published in Wu *et al.* (2011b).

- A new method for adding spatio-temporally coherent millimeter scale surface geometry to coarse dynamic 3D scene models captured from multi-view video under general and unknown illumination (Chapter 5). This is the first method able to capture the true fine dynamic surface detail under general and unknown illumination. The time-varying incident illumination, time-varying and spatially varying surface albedo, and time-varying geometry detail, are reconstructed without using specially engineered and calibrated lights in the scene. The spatio-temporal information in the scene is exploited through soft temporal priors in a maximum a posteriori probability inference framework, which improves reconstruction quality but permits variations in the data. This work has been published in Wu *et al.* (2011a).

- A new theoretical formulation of performance capture that simultaneously recovers human articulated motion, the surface shape and time-varying incident illumination, by minimization of shading-based error (Chapter 6). This method is able to reconstruct both skeletal motion and finely detailed time-varying 3D surface geometry for human performances that are recorded under general and changing illumination and in front of a less constrained background, where previous methods would fail. This work has been published in Wu *et al.* (2012).

- A new passive facial performance capture method that is able to reconstruct high-quality dynamic facial geometry from only a single pair of stereo cameras (Chapter 7). The proposed method achieves detailed and spatio-temporally coherent results for expressive facial motion in both indoor and outdoor scenes, even from low quality input images recorded with a hand-held consumer stereo camera. It is the first method to capture facial performances of such high quality from a single stereo rig. This work has been published in Valgaerts *et al.* (2012b).

- A new performance capture method which is able to capture full body skeletal motion and detailed surface geometry of one or multiple actors using only a single stereo pair of video cameras, which is permitted to move during recording (Chapter 8). It is the first method to apply knowledge about the incident illumination and a detailed spatially-varying BRDF of each actor in a scene for both skeletal pose estimation and for reconstruction of detailed surface geometry. It succeeds under uncontrolled lighting, non-frontal body poses of the actors, scenes in which actors wear general apparel with non-Lambertian reflectance, and it also succeeds in front of general scene backgrounds where classical background subtraction would be infeasible. This work has been published in Wu *et al.* (2013).

1.3 List of Publications

The work presented in this thesis has been published in the following papers:

Wu *et al.* (2011b) **Chenglei Wu**, Bennett Wilburn, Yasuyuki Matsushita, Christian Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 969-976, 2011.

Wu *et al.* (2011a) **Chenglei Wu**, Kiran Varanasi, Yebin Liu, Hans-Peter Seidel, Christian Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In IEEE International Conference on Computer Vision (ICCV), pp. 1108-1115, 2011.

Wu *et al.* (2012) **Chenglei Wu**, Kiran Varanasi, Christian Theobalt. Full body performance capture under uncontrolled and varying illumination: a shading-based approach. European Conference on Computer Vision (ECCV), Part IV, LNCS 7575, pp. 748-761, 2012.

Valgaerts *et al.* (2012b) Levi Valgaerts, **Chenglei Wu**, Andres Bruhn, Hans-Peter Seidel, Christian Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. In ACM Transactions on Graphics (Proc. SIGGRAPH Asia), 31(6), Article 187, 2012.

Wu *et al.* (2013) **Chenglei Wu**, Carsten Stoll, Levi Valgaerts, Christian Theobalt. On-set performance capture of multiple actors with a stereo camera. In

ACM Transactions on Graphics (Proc. SIGGRAPH Asia), 32(6), Article 161, 2013.

Li *et al.* (2013) Guannan Li, **Chenglei Wu**, Carsten Stoll, Yebin Liu, Kiran Varanasi, Qionghai Dai, Christian Theobalt. Capturing relightable human performances under general uncontrolled illumination. In Computer Graphics Forum (Proc. Eurographics), 32(2), pp. 275-284, 2013.

Garrido *et al.* (2013) Pablo Garrido, Levi Valgaerts, **Chenglei Wu**, Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. In ACM Transactions on Graphics (Proc. SIGGRAPH Asia), 32(6), Article 158, 2013.

Chapter 2

Preliminaries

In this chapter we introduce some fundamental concepts for the thesis, including the mathematical description of forward and inverse rendering, the introduction of scene flow and its estimation, the skeleton and the pose parameters, and the surface skinning. Firstly, in Sec. 2.1, we describe the basic equation, i.e. the reflection equation, for rendering a scene, given the illumination, the geometry and the reflectance. Then, in Sec. 2.1.1, we show how to simplify this equation by parameterizing its components with some basis function, i.e. spherical harmonics (SH), and by assuming the reflectance to be Lambertian. In detail, two simplified equations are derived, with applications to two inverse rendering problems, i.e. lighting estimation and geometry estimation. After that, in Sec. 2.1.2 we introduce the generalized form of the SH-parameterized reflection equation, which extends the Lambertian assumption to a more general reflectance function. In Sec. 2.2, scene flow, as well as how to estimate it, are explained. Then, in Sec. 2.3, the skeleton for human motion capture and its pose parameters are explained. Sec. 2.4 introduces the surface skinning.

2.1 Reflection Equation

In order to employ inverse rendering for scene reconstruction, we need to have an understanding of the process of the light transport, namely how images are generated. Fig. 2.1 illustrates a simple example for light transport, where a ray of light hits the surface, gets reflected and is then captured by a camera. Fully realistic images can be synthesized using the rendering equation Kajiya (1986). While it is too complex to directly employ the rendering equation, assumptions can be made to simplify it. By assuming all objects in the scene are non-emitters

2.1 Reflection Equation

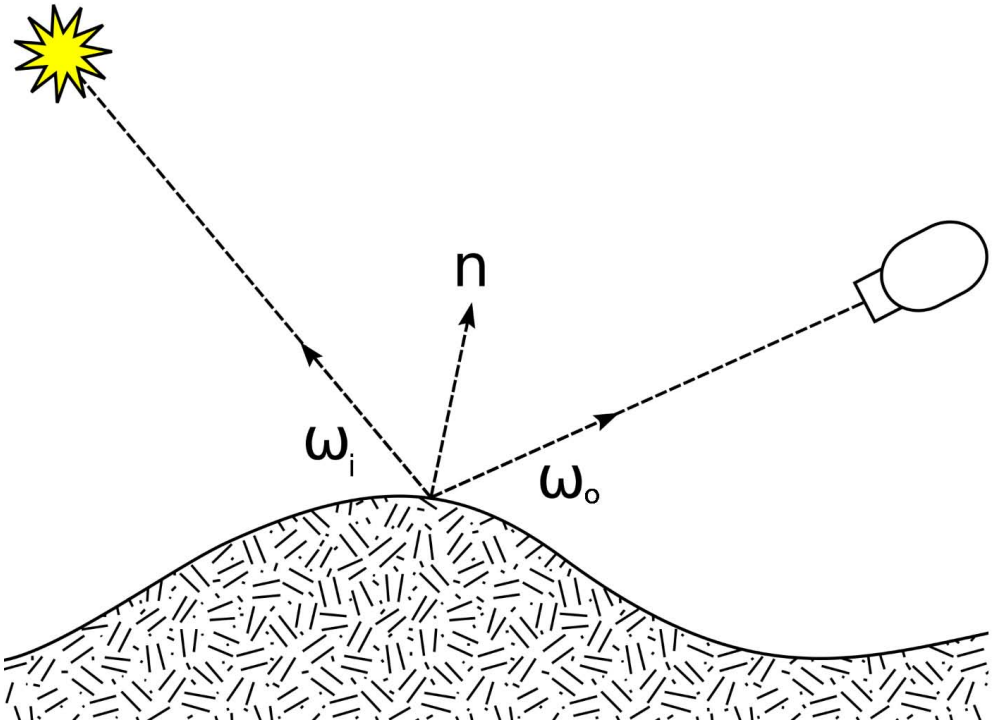


Figure 2.1: An example of light transport [1].

and the light sources are infinitely distant, the light transport can be described by the reflection equation Cohen *et al.* (1993), which is described as:

$$B(\boldsymbol{q}, \boldsymbol{\omega}_o) = \int_{\Omega} L(\boldsymbol{\omega}_i) V(\boldsymbol{q}, \boldsymbol{\omega}_i) \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0) d\boldsymbol{\omega}_i, \tag{2.1}$$

where $B(\boldsymbol{q}, \boldsymbol{\omega}_o)$ is the reflected radiance on the surface point $\boldsymbol{q} \in R^3$, and $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ are the negative incoming light direction and the outgoing direction, both defined in spherical coordinates with respect to the surface normal $\boldsymbol{n}$. The symbol $\boldsymbol{\Omega}$ represents the domain of all possible directions, and $L(\boldsymbol{\omega}_i)$ represents the incident lighting. $V(\boldsymbol{q}, \boldsymbol{\omega}_i)$ is a binary function that defines whether light coming from direction $\boldsymbol{\omega}_i$ is visible by point $\boldsymbol{q}$. $\rho(\boldsymbol{q}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ is the bidirectional reflectance distribution function (BRDF), which defines how light is reflected on the surface and takes the ratio of reflected radiance existing along $\boldsymbol{\omega}_o$ to the irradiance incident on the surface from direction $\boldsymbol{\omega}_i$. A general BRDF usually consists of two components: the diffuse component and the specular component. The diffuse component assumes uniform reflection of the light with no directional dependence.

[1] en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function

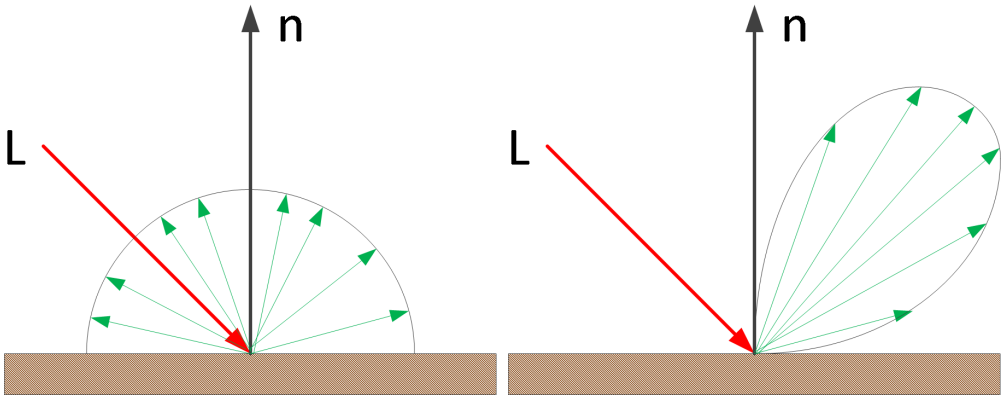


Figure 2.2: Diffuse component and specular component of BRDF. The left image is the illustration of the diffuse component. The right image is the illustration of the specular component. The red line is the incoming light. The blue lines are the reflected irradiance, the length of which describes its intensity. $\boldsymbol{n}$ is the surface normal. While the diffuse component reflects the light uniformly, the reflected irradiance from the specular component is clustered.

The specular component is responsible for view-dependent reflection, e.g. glossy reflection. If defining the incident direction $\boldsymbol{\omega}_i = (\theta_i, \phi_i)$ and outgoing direction $\boldsymbol{\omega}_o = (\theta_o, \phi_o)$, Fig. 2.2 shows an example for the diffuse component and specular component, respectively. Obviously, for the diffuse component, the reflected radiance does not depend on the viewing direction. For the specular component, the reflected radiance changes according to differing viewing angles.

As we are more interested in inverse rendering, the problem here is how to make use of this equation to estimate each component, especially the geometry, from the captured images. However, the integral computation in Eq. (2.1) and the complexity of the BRDF make it prohibitive to directly employ it for inverse rendering. To follow, we will introduce how to simplify the BRDF assumption, and exploit some basis function to circumvent the integral computation.

2.1.1 Lambertian Objects

In order to simplify Eq. (2.1), here we assume the surface reflectance to be diffuse and take Lambert's law to represent the diffuse reflectance, i.e. the BRDF $\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = k_d$, where k_d is a constant value and is called the diffuse albedo. Based on these assumptions, the reflection equation can be simplified Basri & Jacobs (2003); Ramamoorthi & Hanrahan (2001c). Thus, the reflection equation

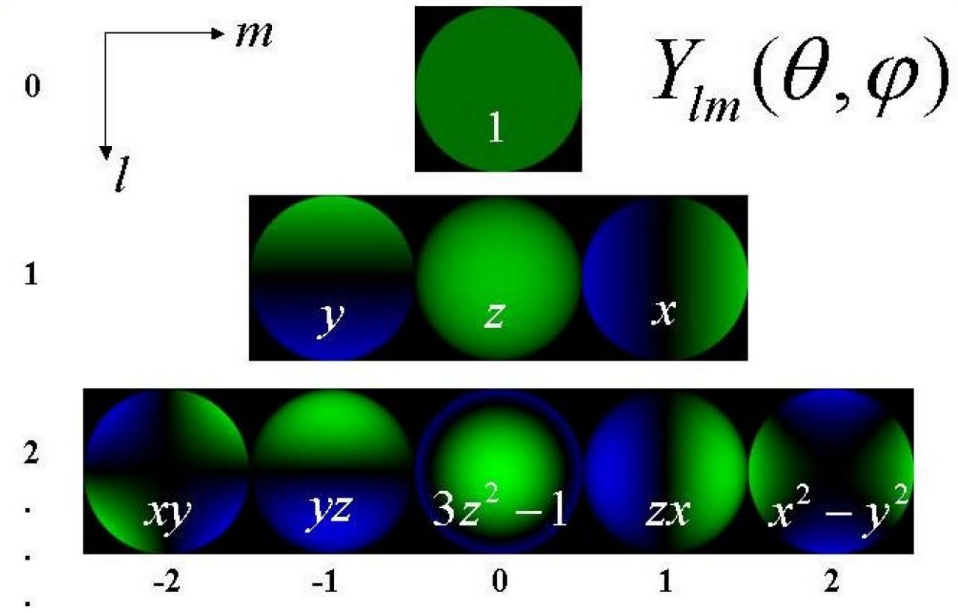


Figure 2.3: The first three orders of real spherical harmonics ($l = 0, 1, 2$) corresponding to a total of 9 basis functions. These images show only the front of the sphere, with green denoting positive values and blue denoting negative values. Ramamoorthi (2005)

for a Lambertian surface is described as:

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \int_{\Omega} L(\boldsymbol{\omega}_i) V(\boldsymbol{q}, \boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0) d\omega_i, \tag{2.2}$$

where the symbols have the same meanings as in Eq. (2.1).

To circumvent the integral computation, one way is to employ orthogonal basis functions to represent each term inside the integral. One naive basis function is the Fourier basis function, but it has been found that the Spherical Harmonics (SH) basis function is more suitable to represent the function that is defined with respect to spherical variables Ramamoorthi & Hanrahan (2004). As in Ramamoorthi & Hanrahan (2001c), we are using the SH representation here. In other words, any function defined in the spherical domain can be represented using a series of SH functions, while the weight for each basis function is called the SH coefficient. The first three orders of SH are shown in Fig. 2.3, where $Y_{lm}(\theta, \phi)$ is the spherical harmonic basis function of the spherical coordinates θ and ϕ. They can also be written as polynomials of the Cartesian components

x, y, z, with $x^2 + y^2 + z^2 = 1$. The indices of the SH function obey $l \geq 0$ and $-l \leq m \leq l$. Thus, there are $2l + 1$ basis functions for a given order l. In order to use the orthogonality of SH, we define $L_v(\boldsymbol{\omega}_i) = L(\boldsymbol{\omega}_i)\, V(\boldsymbol{q}, \boldsymbol{\omega}_i)$ as the visible lighting, rewriting the reflection equation as

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \int_{\Omega} L_v(\boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0) d\boldsymbol{\omega}_i. \tag{2.3}$$

Note that the function $\max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0)$ is rotationally symmetric around the surface normal $\boldsymbol{n}$, and the integral in Eq. (2.3) can be seen as a convolution between the visible lighting term $L_v(\boldsymbol{\omega}_i)$ and the clamped cosine term $\max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0)$. Then, representing both terms with SH, and according to the Funk-Hecke theorem Groemer (1996), the SH coefficients of B can be obtained as

$$B_{lm} = k_d\, g_{lm}\, \hat{\rho}_{dl}, \tag{2.4}$$

where B_{lm}, g_{lm} and $\hat{\rho}_{dl}$ are the SH coefficients of the reflected irrandiance $B(\boldsymbol{q})$, the lighting term and the clamped cosine term. As the clamped cosine term is known, its SH coefficients can be pre-computed. Fig. 2.4 shows the SH coefficients for the first 20 orders. It demonstrates that the coefficients decay very rapidly with increasing the order. From a signal processing perspective, the clamped cosine function acts like a low-pass filter. This means that a low order of SH representation for $B(\boldsymbol{n})$ can achieve a very high representation accuracy, demonstrating the efficiency of employing SH representation for the reflected radiance $B(\boldsymbol{n})$ on Lambertian surfaces. Then, with the SH coefficients B_{lm} known, the reflected radiance $B(\boldsymbol{q})$ can be obtained as

$$B(\alpha, \beta) = k_d \sum_{l=0}^{N_D} \sum_{m=-l}^{l} \Lambda_l\, g_{lm}\, \hat{\rho}_{dl}\, Y_{lm}(\alpha, \beta), \tag{2.5}$$

where (α, β) are the spherical angular parameters of $\boldsymbol{n}$, N_D is the SH order, and Y_{lm} is the SH basis function. Λ_l is a scalar and is defined as

$$\Lambda_l = \sqrt{\frac{4\pi}{2l+1}}. \tag{2.6}$$

As explained, a low order N_D is enough to obtain a high-accuracy representation using SH. Considering that the visible lighting term may have large high-frequency components, we take $N_D = 4$ in this thesis. Eq. (2.5) is much simpler than the original reflection equation and is very favorable for inverse rendering.

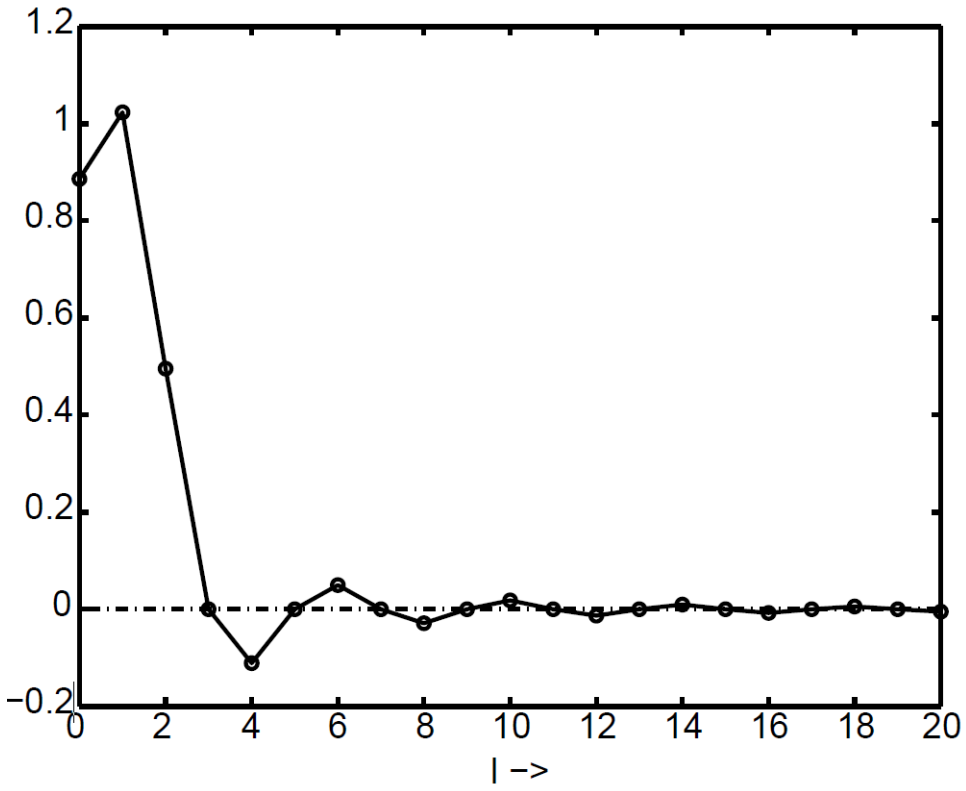


Figure 2.4: SH coefficients of the clamped cosine function. Note that odd terms with $l > 1$ are equal to zero. As l increases, the coefficients decay rapidly. Ramamoorthi (2005)

Given the visible lighting and the captured radiance from the images, the surface normal orientation can be efficiently inferred using this equation. In detail, in Chapters 4, 5, 6, and 7, we employ Eq. (2.5) to inversely estimate the surface normal or the geometry of the scene from image or video input.

Another way to simplify the reflection equation in Eq. (2.2) is to define $T(\boldsymbol{q}, \boldsymbol{\omega}_i) = V(\boldsymbol{q}, \boldsymbol{\omega}_i)\ \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0)$, and to represent $L(\boldsymbol{\omega}_i)$ and $T(\boldsymbol{q}, \boldsymbol{\omega}_i)$ with SH. According to the orthogonality of the SH basis function, the reflection equation becomes

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \sum_{l=0}^{N_D} \sum_{m=-l}^{l} L_{lm}\, T_{lm}(\boldsymbol{q}), \tag{2.7}$$

where L_{lm} and $T_{lm}(\boldsymbol{q})$ are the SH coefficients of $L(\boldsymbol{\omega}_i)$ and $T(\boldsymbol{q}, \boldsymbol{\omega}_i)$. In this equation, all the surface points share the same global lighting environment, which is represented by a set of SH basis functions here. Thus, given the geometry

and taking the captured image radiance as input, the lighting can be inversely estimated using this equation. The inverse lighting using Eq. (2.7) is exploited in Chapters 4, 5, 6, and 7.

2.1.2 General BRDF

General BRDF consists of not only a diffuse part, but also a specular part. As the diffuse part is modeled as Lambertian reflectance, the specular part can be represented by a bunch of different models Ngan *et al.* (2005). In this section, we focus on the specular part of the BRDF, as the irradiance from the diffuse part can be efficiently computed with simplified equations in Sec. 2.1.1. For specular component, the Phong reflectance model Phong (1975) is widely used owing to its simplicity. It is described as follows:

$$\rho_s(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{s+1}{2\pi}(\boldsymbol{r} \cdot \boldsymbol{\omega}_o)^s, \tag{2.8}$$

where s is the shininess value, and $\boldsymbol{r} = 2(\boldsymbol{n} \cdot \boldsymbol{\omega}_i)\boldsymbol{n} - \boldsymbol{\omega}_i$ is the reflected direction of $\boldsymbol{\omega}_i$ about the normal $\boldsymbol{n}$.

Although the Phong model is widely employed in many computer graphics applications, it is not physically accurate. The Torrance-Sparrow model, which is derived by modeling physical reflection on the surface as many microfacet reflections, is more accurate when representing real materials Ngan *et al.* (2005). The Torrance-Sparrow model usually consists of three terms, including the microfacet distribution term, the geometric attenuation term and the Fresnel term. The geometric attenuation term accounts for the self-shadowing due to the microfacets. The Fresnel term describes how much light is reflected and how much is refracted. Here, we ignore the geometric attenuation term and the Fresnel term, and a simplified Torrance-Sparrow model is described as

$$\rho_s\left(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o\right) = \frac{k_s}{4\pi\sigma_b^2 \cos\theta_i \cos\theta_o} \exp\left(-(\theta_h/\sigma_b)^2\right) \quad , \tag{2.9}$$

where k_s is the specular albedo; θ_i, θ_o and θ_h are the incoming light direction, the viewing direction and the half angle (of the angle between the light direction and the viewing direction), all defined with respect to the surface normal; and σ_b is the surface roughness. We employ this simplified Torrance-Sparrow model for the specular component of the BRDF in Chapter 8.

With a general reflectance function, the reflection equation can also be similarly represented with SH. For a general BRDF without any parametric modeling

other than isotropic, the rephrased equation of the reflectance equation has the form

$$B\left(\alpha,\beta,\theta_o,\phi_o\right) = \sum_{l=0}^{F_B}\sum_{m=-l}^{l}\sum_{p=0}^{P_B}\sum_{q=-p}^{p} g_{lm}\,\hat{\rho}_{lpq}\,D^l_{mq}(\alpha)\,e^{Im\beta}\,Y_{pq}(\theta_o,\phi_o) \quad , \tag{2.10}$$

where (α,β) and (θ_o,ϕ_o) are the spherical angular parameters of $\boldsymbol{n}$ and $\boldsymbol{\omega}_o$, F_B and P_B are the SH orders, and L_{lm} and $\hat{\rho}_{lpq}$ are the SH coefficients of $L_v(\boldsymbol{\omega}_i)$ and $\hat{\rho}\left(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o\right)$. $D^l_{mq}(\alpha)$ is a matrix modeling how a spherical harmonic transforms under rotation into direction α, and $Y_{pq}(\theta_o,\phi_o)$ is the SH basis function. Note that while (α,β) is defined in global coordinates, (θ_o,ϕ_o) is defined in local surface coordinates, with the normal direction as north pole.

Eq. (2.10) is much more complicated than Eq. (2.5) due to the complexity of the general isotropic BRDF. Due to the complicated formula of $D^l_{mq}(\alpha)$ Ramamoorthi & Hanrahan (2004), it is still very challenging to apply Eq. (2.10) directly for inverse rendering. However, if the BRDF has a central direction, e.g. the simplified Torrance-Sparrow model, Eq. (2.10) can be further simplified. Specifically, taking the form of Eq. (2.9) for the reflectance function, a rephrased reflection equation in the frequency domain, having a form similar to the Lambertian case, can be derived:

$$B_s(\alpha',\beta') = \sum_{l=0}^{N_S}\sum_{m=-l}^{l} \Lambda_l\,L_{lm}\,\hat{\rho}_{sl}\,Y_{lm}(\alpha',\beta') \quad , \tag{2.11}$$

where $\hat{\rho}_{sl}$ are the SH coefficients of the properly reparameterized BRDF, N_S is the order of SH, and (α',β') is the reparameterized spherical angle of (α,β) with respect to the central direction of BRDF. The SH order in Eq. (2.11) is usually higher than the Lambertian case because the frequency spectral of general BRDF will not always be low-pass. In this thesis, we take $F_S = 10$ and will reduce it accordingly when BRDF parameters can be determined.

2.2 Scene Flow Estimation

Finding the corresponding pixels in multiple frames, which is usually called correspondence finding, is also one of the key problems in performance capture. Based on the photo-consistency constraint, which assumes the correspondences share the same color, optical flow describes a 2D displacement field providing dense correspondences between two images Brox *et al.* (2004); Horn & Schunck (1981).

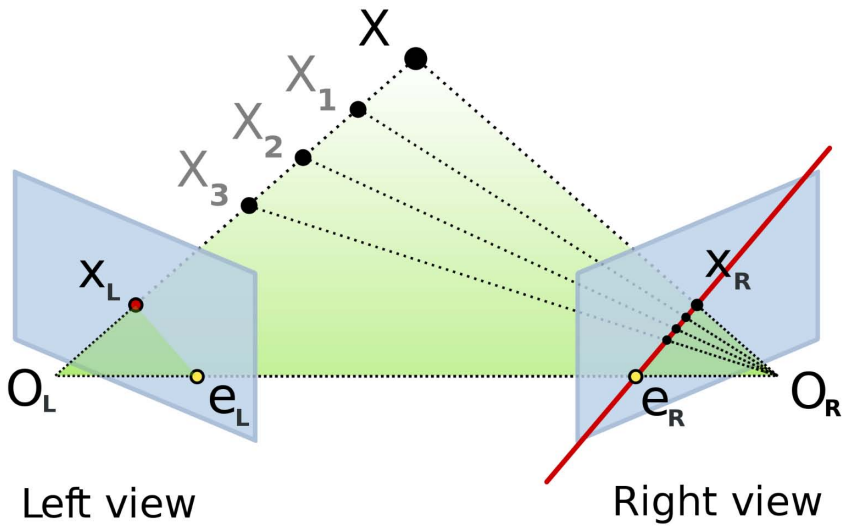


Figure 2.5: Epipolar constraint. O_L and O_R are the two camera centers. x_L and x_R are the projections of the 3D point X in the two cameras. e_L and e_R are the intersections of the baseline O_LO_R with the two camera planes, and are called epipoles. X_1, X_2, X_3 are 3D points lying on the optical ray O_Lx_L. Given point x_L in the left camera, its correspondence x_R in the right camera is constrained to lie on the projection of the optical ray O_Lx_L, i.e. the epipolar line e_Rx_R.

Optical flow is usually employed to capture the 2D motion field between two consecutive frames in a video.

With a stereo camera setting as shown in Fig. 2.5, the corresponding points in the two images cannot lie in arbitrary locations. In fact, they are constrained by the epipolar constraint. In Fig. 2.5, O_L and O_R are the camera center positions for the left and right camera respectively. The point x_R corresponding to the point x_L is actually constrained to lie on a specific line (red line in Fig. 2.5), which is called the epipolar line Hartley & Zisserman (2000). This constraint is called the epipolar constraint, which relates corresponding points in one pair of images by a 3×3 matrix F, i.e. the fundamental matrix. From the estimated correspondences between the left view and the right view, a 3D model can be reconstructed for each frame. With the computed optical flow between two consecutive frames, a 3D motion field can then be obtained; this is called scene flow Vedula *et al.* (2005). Scene flow describes how a surface at the current frame moves to the next frame in 3D. Fig. 2.6 shows an estimated scene flow on a 3D face surface. To follow, we introduce how to estimate the scene flow through a variational framework.

Figure 2.6: Estimated scene flow overlaid with a 3D face surface (red: large motion; blue: small motion).

Fig. 2.7 shows scene flow estimation for two consecutive stereo frames. To compute the scene flow between the time instances t and $t+1$, we employ a scene flow estimation method similar to Valgaerts *et al.* (2010). In contrast to Valgaerts *et al.* (2010), we assume the calibration of the stereo system is known here so we can use the known fundamental matrix to guide the correspondence search.

The scene flow method estimates a 3D reconstruction and 3D displacement field by establishing correspondences in the image domain. It is based on the four frame case depicted in Fig. 2.7. As one can see, all possible constraints between two consecutive stereo pairs (I_0^t, I_1^t) at time t and (I_0^{t+1}, I_1^{t+1}) at time $t+1$ can be expressed in terms of three unknown optical flow fields: the *motion flow* $\boldsymbol{w}_1$, the *stereo flow* $\boldsymbol{w}_2$ and the *difference flow* $\boldsymbol{w}_3$. We compute these flows $\boldsymbol{w}_i = (u_i, v_i)^\top$, $i = 1, 2, 3$, by minimizing an energy function of the form:

$$E = \int_\Omega \Bigg(\underbrace{\sum_{i=1}^{4} E_{\mathrm{D}}^i}_{\text{data}} + \underbrace{\sum_{i=1}^{2} \alpha_i E_{\mathrm{G}}^i}_{\text{geometry}} + \underbrace{\sum_{i=1}^{3} \beta_i E_{\mathrm{S}}^i}_{\text{smoothness}} \Bigg) \, \mathrm{d}\boldsymbol{x} \ . \tag{2.12}$$

The four *data terms* E_{D}^i encode constancy assumptions between all frames, the three *smoothness terms* E_{S}^i assume the desired flows to be piecewise smooth and

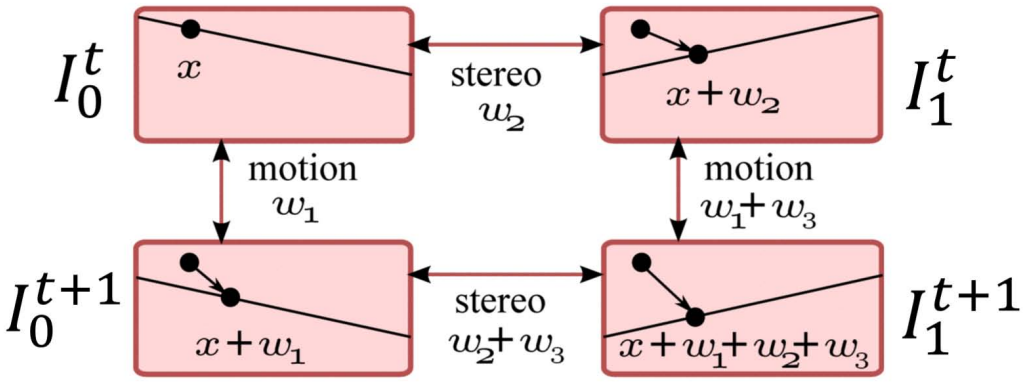


Figure 2.7: Scene flow estimation.

the *geometry terms* E_{G}^i model the geometric relations between the two stereo pairs. All deviations from model assumptions are weighted by positive weights α_i and β_i and are integrated over the rectangular image domain Ω of the reference frame $I_0^t(\boldsymbol{x})$, $\boldsymbol{x} = (x, y)^\top$. Next, we will introduce these terms in detail.

Data Terms For the data constraints that model the relations between the four input images, we first assume that the brightness of corresponding image points is the same in all frames. Using the parameterization of Valgaerts *et al.* (2010) with respect to the coordinates of the reference frame I_0^t, we obtain the four data terms

$$E_{\mathrm{D}}^1 = \Psi\left(|I_0^{t+1}(\boldsymbol{x}+\boldsymbol{w}_1) - I_0^t(\boldsymbol{x})|^2\right), \tag{2.13}$$

$$E_{\mathrm{D}}^2 = \Psi\left(|I_1^{t+1}(\boldsymbol{x}+\boldsymbol{w}_1+\boldsymbol{w}_2+\boldsymbol{w}_3) - I_1^t(\boldsymbol{x}+\boldsymbol{w}_2)|^2\right), \tag{2.14}$$

$$E_{\mathrm{D}}^3 = \Psi\left(|I_1^t(\boldsymbol{x}+\boldsymbol{w}_2) - I_0^t(\boldsymbol{x})|^2\right), \tag{2.15}$$

$$E_{\mathrm{D}}^4 = \Psi\left(|I_1^{t+1}(\boldsymbol{x}+\boldsymbol{w}_1+\boldsymbol{w}_2+\boldsymbol{w}_3) - I_0^{t+1}(\boldsymbol{x}+\boldsymbol{w}_1)|^2\right) \quad . \tag{2.16}$$

While the first two terms result from motion constraints between two consecutive time instances, the last two terms arise from stereo constraints at the same time step. To handle outliers in all constraints independently, every data term is subject to a separate sub-quadratic penalization using the the regularized L_1 norm $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ as the cost function, with $\epsilon = 0.001$. To cope with varying illumination and to make use of color information, we additionally included the gradient constancy assumption in the model and extended it to RGB color images Valgaerts *et al.* (2010).

Geometry Terms The geometric relations between the left and the right image of the stereo pairs (I_0^t, I_1^t) and (I_0^{t+1}, I_1^{t+1}) are given by the associated epipolar constraints. These constraints relate corresponding points in a stereo pair via the fundamental matrix F. The epipolar constraints between the two stereo pairs can be modeled as

$$E_{\mathrm{G}}^1 = \Psi\left(\left((\boldsymbol{x}+\boldsymbol{w}_2)_h^\top F\,(\boldsymbol{x})_h\right)^2\right), \tag{2.17}$$

$$E_{\mathrm{G}}^2 = \Psi\left(\left((\boldsymbol{x}+\boldsymbol{w}_1+\boldsymbol{w}_2+\boldsymbol{w}_3)_h^\top F\,(\boldsymbol{x}+\boldsymbol{w}_1)_h\right)^2\right) \quad , \tag{2.18}$$

where the subscript h denotes the use of homogeneous coordinates, i.e. $(\boldsymbol{x})_h = (x, y, 1)^\top$. In contrast to Valgaerts *et al.* (2010), we assume that the stereo system is calibrated with a known fundamental matrix F. Thus in this case, only the flows $\boldsymbol{w}_i$ are unknown. Both terms E_{G}^1 and E_{G}^2 are soft constraints that penalize deviations of a point from its epipolar line. Together with a sub-quadratic penalizer function such as the regularized L_1 norm (see data terms), such soft constraints increase the robustness of the scene flow estimation with respect to small inaccuracies in the camera calibration.

Smoothness Terms Since the data terms and geometry terms alone may not guarantee a unique solution at every location, the problem needs to be regularized by imposing an additional smoothness constraint. In particular, this makes it possible to obtain dense scene structure and scene flow. In Valgaerts *et al.* (2010), the isotropic total variation (TV) regularizer is used. In our thesis, as we are aiming to capture the geometry of a human face or body, the TV regularizer may not adapt sufficiently to the directional structure, such as laugh lines in a face. Besides, TV can lead to staircasing artifacts, i.e. steps in the reconstructed geometry. To recover the motion of typical facial features more realistically, we need a smoothness constraint that adapts better to the structure of the underlying reference image, while preserving sharp discontinuities in the reconstruction and the scene flow at the same time. Thus, we make use of recent advances in the field of optical flow estimation Sun *et al.* (2008); Zimmer *et al.* (2011) and employ the following anisotropic smoothness term

$$E_{\mathrm{S}}^i \;=\; \Psi_s\left(|\nabla \boldsymbol{w}_i^\top \boldsymbol{r}_1|^2\right) + \Psi_s\left(|\nabla \boldsymbol{w}_i^\top \boldsymbol{r}_2|^2\right) \quad . \tag{2.19}$$

It splits the regularization locally into the directions *along* and *across* the image structures by projecting the Jacobian $\nabla \boldsymbol{w}_i$ onto $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, respectively. Thereby,

the directions $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are computed as eigenvectors of the structure tensor

$$J = K_G * \nabla I_0^t \, \nabla I_0^{t\top} \,, \tag{2.20}$$

where $*$ denotes convolution with a Gaussian K_G. Since deviations from smoothness are penalized separately for each direction, and typically a discontinuity-preserving cost function is used, such as $\Psi_s(s^2) = 2\lambda_s^2\sqrt{1 + s^2/\lambda_s^2}$, with $\lambda_s > 0$, discontinuities in the solution are preserved *independently* for both directions. This in turn is able to handle structures of different intrinsic dimensionality such as corners, edges and homogeneous regions appropriately, thereby achieving the desired *structure-aware* anisotropic smoothing behavior.

Minimization The final energy given in Eq. (2.12) has to be minimized with respect to the three unknown flows $\boldsymbol{w}_i$. To this end, we employ the the minimization scheme from Valgaerts *et al.* (2010): large displacements are resolved by means of a coarse-to-fine multi-resolution strategy, while the resulting nonlinear optimization problem at each resolution level is solved using a bidirectional multigrid method. Please note that in contrast to the original optimization scheme, we do not need to perform an alternating minimization between the flows and fundamental matrix, since F is known here.

With the estimated 2D flow fields, all corresponding pixels are triangulated to obtain a 3D reconstruction and a 3D displacement field, i.e. the scene flow for each reconstructed point. The scene flow estimation is employed in Chapter 7 and Chapter 8 for deformable surface tracking and skeletal motion estimation, respectively.

2.3 Human Skeleton and Pose Parameters

In marker-less full-body performance capture, a prior template with underlying skeletons is frequently used; see Fig. 2.8. This representation is motivated by human anatomy. The full representation of the anatomical bones in a human body is very complex, and it is beyond the realm of possibility to estimate the motion of such a representation. The kinematic skeleton we use is an approximation of a human skeleton where the degrees of freedom (DOF) are reduced to a manageable size. With the skeleton determined by a set of joints and body segments, it has to be determined how the motion parameters on it should be defined. As the motions of body segments depend on each other through the body joints,

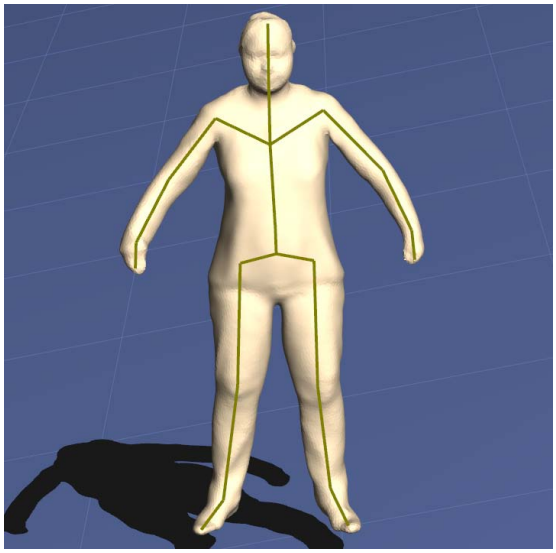


Figure 2.8: Human skeleton.

a convenient way of incorporating these additional constraints is the twist and product of exponentials map formalism for kinematic chains Bregler *et al.* (2004); Murray *et al.* (1994). Using this format, the motion of each body segment can be described as the motion of the previous segment in a kinematic chain and an angular motion around a joint. Just one single DOF for each additional segment in the chain is added. Therefore, the number of free motion parameters can be dramatically reduced using this representation, and the reduced unknown motion parameters will make the motion estimation more robust.

2.3.1 Twist Based Pose Representation

Using the exponential maps, a twist ξ can be represented as (a) a 6D vector, or (b) a 4×4 matrix with the upper 3×3 component as a skew-symmetric matrix:

$$\xi = \begin{pmatrix} \upsilon_i \\ \upsilon_2 \\ \upsilon_3 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}, \hat{\xi} = \begin{pmatrix} 0 & -\omega_z & \omega_y & \upsilon_1 \\ \omega_z & 0 & -\omega_x & \upsilon_2 \\ -\omega_y & \omega_x & 0 & \upsilon_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.21}$$

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ is a 3D unit vector that points in the direction of the rotation axis. The rotation transformation is specified by a scalar angle θ that

is multiplied by the twist: $\xi\theta$. The $\boldsymbol{v} = (v_1, v_2, v_3)$ component determines the location of the rotation axis and the amount of translation along this axis. It can be shown that for any arbitrary rigid motion $G \in SE(3)$ there exists a $\xi \in R^6$ twist representation. A twist can be converted into the G representation with the following exponential map:

$$G = \begin{pmatrix} r_{11} & r_{12} & r_{13} & d_1 \\ r_{21} & r_{22} & r_{23} & d_2 \\ r_{31} & r_{32} & r_{33} & d_3 \end{pmatrix} = \boldsymbol{e}^{\hat{\xi}} = \boldsymbol{I} + \hat{\xi} + \frac{(\hat{\xi})^2}{2!} + \frac{(\hat{\xi})^3}{3!} + \dots \tag{2.22}$$

Kinematic Chain as a Product of Exponentials If we have a chain of $K+1$ segments linked with K joints (kinematic chain) and describe each joint by a twist ξ_k, a point on segment k is mapped by the transformation defined as

$$g_k(\Delta\hat{\xi}, \theta_1, \theta_2, \dots, \theta_k) = e^{\Delta\hat{\xi}} \prod_{i=1}^{k} e^{\hat{\xi}_i \cdot \theta_i}, \tag{2.23}$$

where $\Delta\hat{\xi}$ describes the rigid motion of the root joint, and $\theta_1, \theta_2, \dots, \theta_k$ represent the rotation of each joint (here for simplicity, we just assume one rotation of DOF for each joint). As the human skeleton is a kinematic chain, the skeletal pose of a human can also be represented in the same way. We use the twist based pose representation for human skeletal motion estimation in Chapter 6 and Chapter 8.

2.4 Surface Skinning

Skinning is the process of attaching a renderable skin, e.g. a mesh surface, to an underlying articulated skeleton. This technique is extensively used for animating articulated characters such as virtual humans in computer graphics and interactive applications. In our marker-less motion capture algorithms, we use this technique to deform the template mesh according to the given pose parameters. In order to perform surface skinning, a static character model with an underlying skeleton in a neutral pose is given. A set of blending weights are assigned to each vertex to define the amount of influence coming from different joints.

To introduce different skinning methods, let us assume there are m joints in the model, and that vertex $\boldsymbol{q}$ on the mesh surface is attached to joints $J_1, ..., J_m$ with weights $(w_1, ..., w_m)$. The weights are normally assumed to be convex, i.e., $w_i \geq 0$ and $\sum_{i=1}^{m} w_i = 1$. The blending weight w_i represents the amount of influence of joint J_i on vertex $\boldsymbol{q}$. In the neutral pose, each joint has an associated

local coordinate system. Then, the transformation from the neutral pose of joint J to its actual position in the animated pose can be expressed by a rigid transformation matrix, denoted as C_j. Based on these inputs, the skinning algorithm then solves for the new position of the mesh surface, i.e., a new vertex position $\boldsymbol{q}'$ for each vertex $\boldsymbol{q}$.

2.4.1 Linear Blend Skinning

For linear blend skinning, each neutral pose vertex is firstly rigidly transformed by all of its influencing joints. Then, blending weights are used to linearly combine these transformed positions into one position. Mathematically, the updated vertex position is given as

$$\begin{pmatrix} \boldsymbol{q}' \\ 1 \end{pmatrix} = \sum_{i=1}^{m} w_i C_{J_i} \begin{pmatrix} \boldsymbol{q} \\ 1 \end{pmatrix}, \tag{2.24}$$

where C_{j_i} represents the rigid transform matrix for joint J_i, $\boldsymbol{q}$ and $\boldsymbol{q}'$ are the vertex positions before and after skinning. Linear blend skinning is used in the skeletal motion estimation in Chapters 6 and 8.

2.4.2 Dual Quaternion Skinning

Unfortunately, linear blend skinning is known to suffer from skin collapsing artifacts, as the blended matrix $\sum_{i=1}^{m} w_i C_{J_i}$ is no longer a rigid transformation. Thus, Kavan *et al.* (2007) propose a new blending method based on dual quaternions, which is called dual quaternion skinning. This method first converts the rigid transformation matrices $C_{J_i}, ..., C_{J_m}$ to unit dual quaternions $Q_1, ..., Q_m$. Then, a blended unit dual quaternion Q_b w.r.t. the given blending weights $(w_1, ..., w_m)$ is computed using a linear combination and then a normalization:

$$Q = \frac{w_1 Q_1 + \cdots + w_m Q_m}{\|w_1 Q_1 + \cdots + w_m Q_m\|}. \tag{2.25}$$

Finally, the blended dual quaternion Q is converted back to a rigid transformation matrix M. The updated vertex position is computed as

$$\begin{pmatrix} \boldsymbol{q}' \\ 1 \end{pmatrix} = M \begin{pmatrix} \boldsymbol{q} \\ 1 \end{pmatrix}. \tag{2.26}$$

As M is assured to be a rigid transformation, the skin collapsing is prevented.

Chapter 3

Related Work

In this chapter, we introduce the previous work related to the content in the thesis. It is generally divided into three areas. Firstly, we introduce the related work in image based modeling for static objects. This is related to our work in Chapter 4, which focuses on static 3D reconstruction. Secondly, the work related to performance capture, including full body capture and face capture, is discussed. This section introduces the work respectively related to Chapters 5, 6 and 8 for capturing full-body performance, and Chapter 7 for capturing facial performance. In the third part, the related work in the field of reflectance estimation and lighting estimation is introduced. Reflectance estimation and lighting estimation are two basic techniques in inverse rendering, and thus related to the content of the thesis as a whole.

3.1 Static 3D Reconstruction

Stereo matching is one of the basic techniques in computer vision to estimate the 3D structure, e.g. the depth, from one pair of images. The basic idea is to estimate the correspondence between two images based on the photo consistency constraint, e.g. requiring the color of the correspondences in two images to be the same. Then, the depth can be triangulated from the correspondences. Multi-view stereo (MVS) extends the stereo method into working with multi-view images, and is able to reconstruct watertight 3D geometry. This technique has achieved great success in static 3D reconstruction. The reconstruction accuracy of the most advanced MVS methods is around 1/400 (0.5mm for a 20 cm wide object) Seitz *et al.* (2006). These techniques can be generally divided into two categories. The first group is formed by multistage local approaches, which proceeds the

reconstruction stage by stage, for instance, first estimating the depth maps for each camera and then fusing them to a 3D model Bradley *et al.* (2008); Liu *et al.* (2010), or first reconstructing the 3D points for extracted features and then growing them into a watertight 3D model Furukawa & Ponce (2010). The other group is global methods, which formulate the 3D reconstruction as a global optimization, e.g. extracting the surface from a volume representation Vogiatzis *et al.* (2007), or evolving a surface by optimization Pons *et al.* (2005). In spite of its great success in 3D reconstruction, this technique has an inherent weakness in reconstructing high-frequency surfaces.

Shape-from-shading (SfS) is another important technique for image based modeling. SfS can be seen as one of the oldest inverse rendering techniques, which tries to estimate the shape by inverting the rendering process from a single image. Traditionally, a surface normal map is first estimated and then integrated to obtain the shape. The history of SfS can be traced back to the 1970s Horn (1970). In terms of the traditional SfS work, we refer the reader to Zhang *et al.* (1999) for a survey. As only one single image is available and the unknowns in lighting, reflectance and shape are too many to solve, the SfS problem is not a well-posed problem. Traditional SfS usually assumes a single known light source and uniform reflectance to make the problem solvable. Many priors, e.g. the surface integrability, are also employed to better constrain the solution. Even though great advances have been made in the last decades, SfS has yet to find its way into real-world applications. Alternatively, based on the shading cues but using the images filmed under many different lightings, photometric stereo (PS) methods have recently attracted much attention within the field Woodham (1980). As a set of images with different lightings is taken as input, the shape estimation problem becomes well-posed and high-quality shapes can be estimated. Specifically, Alldrin *et al.* (2008) propose a new photometric stereo method which works for isotropic BRDF. For uncalibrated photometric stereo, with a point light source assumption, the shape can be solved up to a transformation, which is called generalized bas-relief ambiguity (GBR) Belhumeur *et al.* (1997). Shi *et al.* (2010) develop a self-calibrated method for photometric stereo, even when the camera response function is unknown. Under general lighting conditions, where the lighting is not just generated simply by point lights but also produced by other types of lights as well as the surrounding environment, another photometric stereo method is proposed for Lambertian surfaces by assuming no cast shadows in the scene Basri *et al.* (2006). Although high-quality shapes can be obtained, there are still many constraints to applying photometric stereo methods. For

instance, an image sequence must be captured under several different lightings, which may require a special setup for the capture. Besides, because all the images are captured from a single viewpoint, the shape obtained by photometric stereo is not metrically meaningful, even though fine details can be captured. This is also true with the SfS techniques.

From the above analysis, a 3D model can be robustly reconstructed by MVS, but may lack details, while the shading-based approaches (SfS or PS) are quite successful for estimating the high-frequency geometric detail but have difficulty in obtaining metrically correct shapes. It is obvious that these two techniques are complementary and should be combined Blake *et al.* (1986). Much work has been done on efficiently combining them. Leclerc & Bobick (1991) use stereo to provide initialization and boundary constraints for SfS. Cryer *et al.* (1995) combine depth maps from SfS and stereo in the frequency domain using filtering. Rather than fusing MVS and SfS results, Fua & Leclerc (1995) start with a coarse mesh computed from binocular or tri-view stereo, then minimize an error function with stereo, shading, and smoothness components. They handle slowly varying albedo of Lambertian surfaces. Samaras *et al.* (2000) iteratively estimate both shape and illumination given multiple views taken under fixed illumination. They assume piecewise constant albedo. Jin *et al.* (2000, 2004a,b) have proposed a series of variational algorithms that combine MVS and SfS. Their recent work Jin *et al.* (2008) focuses on 3D reconstruction of Lambertian objects with piecewise constant albedo. Hernandez *et al.* (2008) develop a system to capture multi-view images under different lightings. Wu *et al.* (2010) develop a multi-view and multi-lighting system to combine MVS and PS under general lightings for high-quality 3D reconstruction. Beeler *et al.* (2010) recently propose a high-quality stereo method with an additional step of shape refinement. Their refinement embosses or extrudes the geometry at locations where high-frequency shading variations are visible, producing qualitatively pleasing results. However, their strategy for shading-based refinement implicitly requires uniform lighting to work, which prevents the method from being applied to more general illumination conditions. In conclusion, none of those methods is able to work under a common condition in the real world, where the lighting is general, unknown and mostly constant. In Chapter 4 of this thesis, we propose such a technique, which combines MVS and SfS to achieve high-quality 3D geometry by using the images captured under general, unknown and constant lighting.

3.2 Dynamic Scene Reconstruction

3.2.1 Full Body Capture

Marker-less full body motion capture approaches reconstruct human skeletal motion and have been developed in vision and graphics over many years. For a thorough discussion and a historical perspective on this technique, one should consult any of the surveys Moeslund *et al.* (2006); Poppe (2007); Sigal *et al.* (2010). Research efforts today can be broadly distinguished into quality-oriented methods which usually employ multiple synchronized and calibrated cameras to achieve a high level of accuracy, and general purpose methods that work under fewer cameras in potentially cluttered surroundings — albeit producing pose estimates of lower accuracy. Many of the successful methods Bo & Sminchisescu (2010); Lee & Elgammal (2010); Li *et al.* (2010) validated on the HumanEva dataset Sigal *et al.* (2010) rely on a set of training poses for tracking which limits their generalizability to new poses not observed in the training set. For the first category, which is related to Chapters 6 and 8, most of the methods rely on a template skeleton with attached shape template, to then minimize some form of model-to-image consistency, e.g., edge or silhouette features using local or global optimization methods Bregler *et al.* (2004); Deutscher *et al.* (2000); Gall *et al.* (2008). Recently, Stoll *et al.* (2011) have proposed a technique which approaches real-time performance and captures complex motion. Even though these motion capture algorithms estimate the skeletal motion, they do not reconstruct detailed surface models. To move towards combined skeleton and surface capture, researchers experimented with coarse 3D shape models in multi-view motion capture Balan *et al.* (2007), e.g., with parametric human templates. However, they deliver very coarse geometry and expect actors to wear skin-tight clothing. There are also approaches trying to estimate skeletal motion from monocular input, but requiring heavy manual interaction, e.g. Wei & Chai (2010).

Marker-less performance capture approaches go beyond motion capture and reconstruct dynamic geometry, possibly with skeletal motion, of people in more general clothing. Some techniques rely on shape-from-silhouette or active or passive stereo Matusik *et al.* (2000); Starck & Hilton (2007); Waschbüsch *et al.* (2005); Zitnick *et al.* (2004). Vlasic *et al.* (2009) record a person with multiple cameras in a dense controlled light stage and perform photometric stereo for capturing space-time-incoherent shapes. Model-based approaches deform a shape template such that it resembles a person de Aguiar *et al.* (2008); Gall *et al.* (2009);

Vlasic *et al.* (2008) in multi-view video, which yields spatio-temporally coherent reconstructions. Mesh-based tracking approaches, as proposed by de Aguiar *et al.* (2008), provide frame-to-frame correspondences with a consistent topology. The approach by Cagniart *et al.* (2010a) makes a weaker a priori assumption by modeling the scene as a set of moving patches that are tracked over time. But the reconstructed detail of geometry is limited to the patch size. Another set of model-based approaches combine skeleton tracking with deformable surface tracking to capture people in more general apparel Gall *et al.* (2009); Liu *et al.* (2011); Vlasic *et al.* (2008). Some of these methods combine pose estimation with image segmentation and optical flow Bray *et al.* (2006); Brox *et al.* (2006, 2010), and by this means also capture more than one person in a scene Liu *et al.* (2011). However, most methods are still restricted to controlled studios with green screen backgrounds, and usually expect ten or more cameras. Moreover, the amount of surface detail captured by these approaches is limited. Hasler *et al.* (2009) jointly employ feature-based performance capture and structure-from-motion of the background for outdoor motion capture with multiple cameras, but require manual interactions and do not produce detailed surface geometry. Besides, none of these methods are able to work under varying lighting conditions, as the employed image cues will become unstable under changing illuminations. In contrast, in Chapter 6 of this thesis, by employing the shading cues and modeling the lighting changes, we are able to capture both the skeletal motion and the detailed surface geometry under general and varying lighting conditions. Another limitation with the previous marker-less performance capture methods is the requirement for a multiple camera setup. In Chapter 8, we take a step further by just employing a stereo camera setup to track skeletal motion and detailed space-time coherent surface geometry.

There are also recent works on skeletal pose estimation from depth cameras, such as the Kinect, e.g., Ganapathi *et al.* (2010); Shotton *et al.* (2011); Wei *et al.* (2012). These approaches are designed for real-time use and reconstruct coarse skeletal motion and coarse surface geometry Taylor *et al.* (2012). High-quality pose and shape reconstruction is not their goal. In addition, most depth cameras only work indoors, and have a very limited range and accuracy. Besides, some earlier vision methods attempted to capture human skeletal motion from stereo footage, e.g., Plankers & Fua (2001), but did not achieve as high-quality poses and reconstructions as recent methods.

To reconstruct fine scale surface detail, a controlled light setup is usually employed. For instance, Vlasic *et al.* (2009) uses a complex controlled light stage

to achieve a high quality dynamic scene reconstruction from multi-view video. From a single camera, photometric stereo methods are developed to estimate the dynamic surface orientation. Brostow *et al.* (2011) developed a system to capture the dynamic scene under controlled and colored lights. Then, photometric stereo can be utilized to estimate the surface normal for each frame, as each color channel provides one lighting condition. Kim *et al.* (2010) propose another system with a colored lights setup, and also leverage the temporal sample for applying photometric stereo. However, these methods all need a sophisticated controlled light setup, and cannot be applied to general capture conditions. In comparison, in Chapter 5, we propose a dynamic shape refinement method, which is able to capture fine scale geometric detail under general, unknown and uncontrolled illumination. This method is extended in Chapter 8 to even capture the geometric detail on non-Lambertian surfaces.

Instead of making priori template assumptions, some approaches build up a spatio-temporally coherent shape model by space-time analysis of partial scanner data Liao *et al.* (2009); Tevs *et al.* (2012). For single objects in a scene, these approaches also succeed with sparse depth camera or scanner systems. The quality of these methods heavily depends on the quality of the scan or depth data. Besides, it may be infeasible to build such a system in a general environment. Furthermore, due to strong regularization employed in the algorithms, they often capture geometry lacking high-frequency detail. In contrast, in Chapter 8, using a sparse camera system, i.e. a stereo setup, our method is able to achieve detailed reconstructions in a general environment.

3.2.2 Face Capture

Another important branch of performance capture is to capture the dynamic shape of facial expressions, which is called facial performance capture. As the surface of a dynamic face is highly non-rigid, it cannot be easily represented using a skeleton-based model as a human model. Thus, the methods for full-body performance capture cannot be easily extended to facial performance capture. In this section, we review the previous work on capturing facial performances, which is related to Chapter 7 in the thesis.

For many years, researchers in graphics and vision have investigated facial performance capture approaches that differ in the employed sensors and reconstruction techniques. Some methods solely rely on dynamic 3D shape scanner data, i.e., time-varying point clouds, and no additional input images. Anuar

and Guskov Anuar & Guskov (2004) track an initial template mesh from point cloud data using a purely geometric 3D scene flow method. Reconstruction of high frequency detail is difficult with their approach and the purely geometric 3D scene flow method more frequently suffers from drift. Wand *et al.* (2009) simultaneously build up and track a template of a face from point cloud data, but reconstructions lack some high-frequency shape detail. Popa *et al.* (2010) propose a similar framework that can capture more high-frequency detail by means of a change prior. But the detail does not truly come from the capture. Weise *et al.* (2011) use point clouds from a Kinect and a template with an attached blend shape model to track facial performances. However, their goal is animation transfer, not authentic reconstruction of fine-scale shape detail.

Image-based approaches help to overcome the resolution limits and the limits in tracking accuracy that purely geometric methods still have. Following the marker-based motion capture paradigm widely accepted in industry, researchers attempted to reconstruct facial performances by tracking attached or painted markers in a face with several cameras, or by tracking the distortion of an invisible paint applied to the skin Bickel *et al.* (2007); Furukawa & Ponce (2009); Guenter *et al.* (1998); Williams (1990). Active fiducials greatly enhance tracking accuracy and enable robust reconstruction of even extreme facial expressions. However, the resolution of the captured geometry is limited, the mark-up phase can be cumbersome, and due to the active intrusion into the scene, the simultaneous reconstruction of geometry and appearance is not feasible. Huang *et al.* (2011) try to overcome some of these limitations in a data-driven way by transferring geometric detail from a sparse set of 3D scans to dynamic face geometry recorded with a marker-based motion capture system.

Instead of markers, active illumination, e.g., patterns emitted from projectors, can be used to facilitate image-based face geometry reconstruction from multiple cameras Wang *et al.* (2004); Weise *et al.* (2007); Zhang *et al.* (2004). With these approaches, texture acquisition requires interleaving of pattern and texture frames, and temporal reconstruction artifacts may occur since several subsequent images are required for a single 3D reconstruction. Also, establishing geometric correspondence between subsequent reconstructions is still a challenge.

Template-based methods fit a deformable shape model to images of a face Blanz *et al.* (2003); DeCarlo & Metaxas (1996); Pighin *et al.* (1999). While this yields spatio-temporally coherent reconstructions, the captured face geometry is often coarse and lacks true fine-scale detail.

3.2 Dynamic Scene Reconstruction

Our method in Chapter 7 falls into the passive facial performance capture category. High quality facial performances can be reconstructed with purely passive stereo-based approaches in combination with mesh tracking Bradley *et al.* (2010). Borshukov *et al.* (2003) developed the Universal Capture system for the movie *The Matrix*, which deforms a laser-scanned 3D facial model by using optical flow fields computed from a multi-camera system. These approaches usually require dense multi-camera setups and a controlled studio environment. Also, reconstructing pore-level detail is difficult from pure stereo, and temporal drift in the reconstructions often prevents capturing expressive facial motions. Beeler *et al.* (2011) try to overcome the drift problem in dense multi-view face reconstruction by stabilizing mesh tracking with a set of key facial poses. The commercial system by DepthAnalysis[1] also reportedly uses stereo reconstruction from a dense multi-camera system under controlled studio lighting. Stereo reconstruction is also used in the MOVA Contour system[2], which employs an even denser array of tens of cameras, and invisible make-up to aid reconstruction. In contrast to the requirement for a multiple camera setup, the method proposed in Chapter 7 only needs two cameras, i.e. a simple stereo setup, but is able to reconstruct high-quality facial geometry.

Passive acquisition of true fine-scale surface detail with image-based methods is still difficult. Several approaches have recently shown that shading and reflectance effects under controlled lighting can boost reconstruction resolution dramatically. Vogiatzis & Hernández (2011) use controlled tri-colored studio illumination and a combination of multi-view stereo and photometric stereo to capture facial geometry. Combining active structured light scanning and marker-based facial performance capture with a complex light stage illumination setup also enables high-quality capture of geometry and appearance in a studio Alexander *et al.* (2009). Light stage illumination requires recording of multi-view images under several light conditions to obtain a single reconstruction. To cope with the resulting spatio-temporal alignment problem in the data, Wilson *et al.* (2010) developed an approach to establish correspondences between images taken under starkly varying spherical gradient illuminations from the light stage. This enables a combination of stereo and photometric normal reconstruction in a spatio-temporal way. Fyffe *et al.* (2011) reconstruct geometry and reflectance of a moving face using spherical gradient illumination from a light stage and high-speed cameras. However, the reconstructed geometry is not spatio-temporally coherent.

[1] www.depthanalysis.com
[2] www.mova.com

Compared with those approaches, our method in Chapter 7 does not require those specifically engineered setups and is able to achieve highly detailed and spatio-temporally coherent facial performance capture in general and uncontrolled or even changing lighting scenarios.

3.3 Reflectance and Lighting Estimation

Reflectance and lighting are two basic elements in the rendering procedure. In the field of inverse rendering, many methods are developed to estimate reflectance or lighting directly from the captured images. As this thesis is about exploiting inverse rendering for scene reconstruction, the previous work on reflectance and lighting estimation is generally related to all the chapters in the thesis.

In the past, a variety of approaches have been proposed for image-based estimation of reflectance models for static scenes. Having a shape model, samples of surface reflectance can be recorded by capturing images of the object from varying outgoing and incident light directions with a calibrated point light. An analytical model of surface reflectance, such as a parametric BRDF, can now be estimated for the whole surface or for every surface point individually, e.g. Lensch *et al.* (2003); Matusik *et al.* (2003); Sato *et al.* (1997). Given a shape model and some general prior assumptions about lighting Yu & Malik (1998), or given geometry and calibrated lighting Yu *et al.* (1999), the spatially-varying BRDF of a scene can be found via inverse global illumination. Given a manually designed model of the geometry and lighting, BRDF estimation from a single image is feasible Boivin & Gagalowicz (2001). Theobalt *et al.* (2007) extend this concept to scenes with a moving human. They reconstruct a shape and motion of the actor using a template-based motion estimation approach from multi-view video recorded under the light of two calibrated spotlights. From the data, they estimate a parametric BRDF model for each vertex. However, their 3D models are very coarse, which has a negative influence on the final result. Using wavelet-based lighting and an assumed subspace of BRDFs, the surface reflectance of a static object can be estimated using images from community image databases Haber *et al.* (2009).

Another important component in inverse rendering is to estimate the incident illumination from images. For this purpose, certain assumptions are made about the lighting model. A simple point light source assumption is common in uncalibrated photometric stereo methods Higo *et al.* (2009). However, this assumption is too simple to model the real-world illumination. General illumination can be represented by an environment map Greene (1986). Different methods are

developed to parameterize the environment map of incident illumination. Low-frequency illumination can be efficiently represented using Spherical Harmonics (SH), as was shown by Ramamoorthi & Hanrahan (2001a). The SH basis has also been used in the signal processing theory of inverse rendering Ramamoorthi & Hanrahan (2001c). The general illuminations are estimated jointly with a surface normal by ignoring the cast shadow Basri *et al.* (2006). In addition to spherical harmonics, Ng *et al.* (2004) proposed using a Haar wavelet basis to model high-frequency lighting and reflectance effects. The wavelet-based lighting is estimated from community image databases using the estimated BRDF and the reconstructed geometry Haber *et al.* (2009). However, none of these methods consider the light visibility in their work, while in this thesis the light visibility is explicitly modeled.

Reflectance and lighting estimation in this thesis are also mainly considered in the context of shape and motion estimation. Many photometric stereo methods fall into this category. Georghiades (2003) use photometric stereo to capture static shape and BRDF of a face from multiple images illuminated with a point light from unknown directions. Goldman *et al.* (2005) reconstruct shape and spatially-varying BRDF via photometric stereo from images under controlled lighting. However, as mentioned before, photometric stereo is not able to reconstruct metrically correct geometry. Using multi-view images under known illumination, Yoon *et al.* (2010) estimate a parametric BRDF model and exploit this for surface refinement. But it assumes a simple point light source with known position and intensity, and only captures static objects. Carceroni & Kutulakos (2002) capture coarse surfel-based geometry and reflectance from multi-view video footage with calibrated lights. Georghiades (2003) reconstruct a static face model and estimate a coarse BRDF from multiple images under point light illumination with unknown positions. The reflectance and lighting estimation in this thesis differs from those methods in that we are estimating the reflectance and lighting from images captured under one general and uncontrolled illumination, and using the estimated reflectance and lighting for shape and motion estimation.

Part I

Static 3D Reconstruction from Multi-view Images

Reconstructing the 3D geometry of static scenes from multi-view images has been a subject of research for decades. Multi-view stereo methods have achieved great success and are able to reconstruct the geometry of scenes by matching the correspondences across different viewpoints. However, these methods are not able to reconstruct the high-frequency geometric detail. Shading cues can be combined to achieve high-quality reconstruction, but the way shading information is utilized usually assumes a simple point light source, and usually the light is calibrated. These requirements prevent these methods from being applied to general scenarios, where the lighting is general and unknown, e.g. an indoor environment.

In this part, by investigating the inverse of the reflection equation introduced in Chapter 2, we are able to achieve high-quality 3D reconstruction in the general scenario, where the lighting is general and unknown. By inferring the lighting and geometry from images captured under general illumination, the shading cues can be integrated into the reconstruction method to achieve high-quality geometry, which rivals laser scan results. The success of our method on static scene reconstruction proves the concept in this thesis that inverse rendering can be employed to achieve high-quality scene reconstruction.

Chapter 4

High-quality Shape from Multi-view Stereo and Shading under General Illumination

4.1 Introduction

Multi-view stereo (MVS) methods compute depth by triangulation from corresponding views of the same scene point in multiple images. Establishing correspondence is difficult within smoothly shaded regions, so MVS methods compute accurate depth for a sparse set of well-localized points and must interpolate elsewhere. Seitz *et al.* (2006) present a taxonomy and evaluation of MVS algorithms. Results posted on the benchmark website[1] accompanying that work show that today's best-performing methods capture the rough shape of the scene well, but generally cannot recover the high-frequency shape detail well. In contrast to MVS, shape-from-shading (SfS) methods compute per-pixel surface orientation instead of sparse depth. SfS techniques use shading cues to estimate shape from a single image, usually taken under illumination from a single direction Zhang *et al.* (1999). It has been shown that SfS approaches are able to recover high-frequency shape detail, even if surfaces are smoothly shaded. SfS reconstruction can therefore often shine where stereo fails, and vice versa. Generalizing this shading-based reconstruction to the multi-view case is not easy, though. Recovered normal fields usually need to be integrated to obtain 3D geometry, which is non-trivial for general surfaces seen from multiple viewpoints Nehab *et al.* (2005).

[1] vision.middlebury.edu/mview/

Furthermore, most SfS algorithms make strong assumptions about the incident illumination, which effectively restricts most of them to studio lighting conditions.

In this chapter, we re-visit the 3D reconstruction problem from the perspective of inverse rendering. Image-based 3D reconstruction algorithms usually try to exploit some image cues to infer the 3D geometry. What image cues are good for 3D reconstruction? In order to answer this, we need to first understand how the images are generated from the real world. The rendering technique tells us that the images are determined by the lighting, the surface reflectance and the geometry, while the generation process can be described by the rendering equation Kajiya (1986). Thus, the intuitive way to do 3D reconstruction is to invert this process to estimate the geometry, which means solving an inverse rendering problem. From this perspective, SfS is actually one form of inverse rendering technique. With this concept in mind, it is natural to develop a new SfS technique which works for general scenarios, as we understand well how the images are generated under general illumination. However, directly decomposing images into lighting, reflectance and geometry is too ill-posed a problem to be solvable. In our method, we make an assumption about the reflectance, namely that it is Lambertian reflectance with uniform albedo, and start the geometry with the MVS result. With this, we are able to solve two inverse rendering problems, including the lighting estimation and shape refinement, and thus obtain a high-quality 3D reconstruction. This algorithm is one example of exploiting inverse rendering concept for scene reconstruction.

Specifically, in this chapter, we propose a new multi-view reconstruction approach that combines the strengths of MVS and SfS. It enables us to capture high-quality 3D geometry of Lambertian objects from images recorded under fixed but otherwise general, unknown illumination. In detail, we propose a shape reconstruction method that uses stereo for initial geometry estimation and shading-based shape refinement under general and uncalibrated illumination. Our method estimates high-fidelity shapes that include subtle geometric details that cannot be captured by triangulation-based approaches. We develop a new multi-view shading constraint for achieving this goal. For efficient computation, we use spherical harmonics (SH) to estimate and encode general lighting conditions and local visibility. We also develop an adaptive anisotropic smoothness term for preserving high-frequency details while filtering out noise. In addition, we show an adaptive computation approach that takes the complexity of lighting and visibility estimates into account at each surface point for efficient computation. The work presented here was published in Wu *et al.* (2011b).

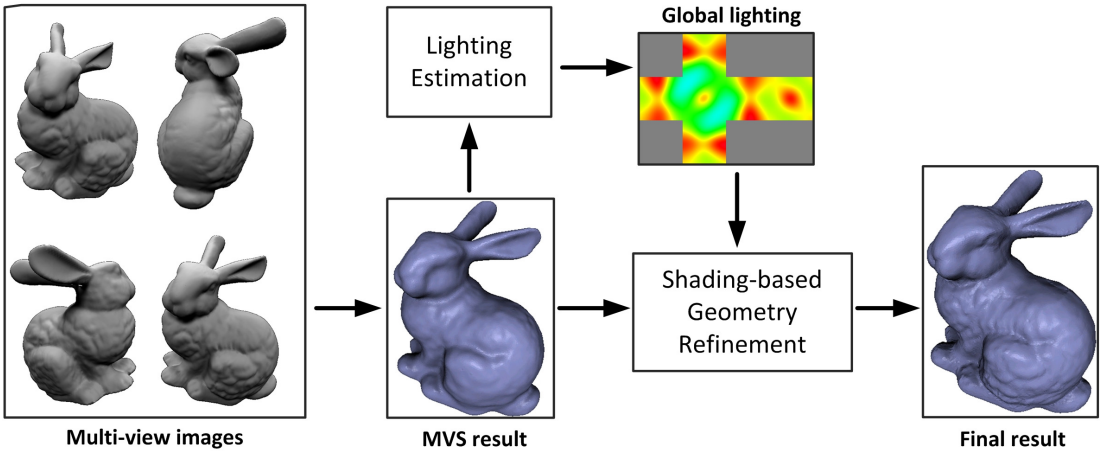


Figure 4.1: Outline of our processing pipeline.

4.2 Method Overview

Our goal in this chapter is to compute a high-quality shape of a static object based on multiple images taken from different viewpoints. The illumination is assumed to be fixed and distant, but is otherwise general and unknown. Cameras are assumed to be calibrated both geometrically and radiometrically. We represent the geometry using a high-resolution mesh model and the illumination using spherical harmonics. In order to keep the problem tractable, we henceforth assume that the albedo of the object is constant. We will see in our results, though, that this assumption does not prevent us from reconstructing detailed shape models even in the presence of small albedo variations. We also neglect inter-reflections on the object.

The workflow of our method is shown in Fig. 4.1. It has three steps. First, we use existing MVS methods to create an initial closed, 3D triangle mesh model of the object. Next, we use this model to estimate the spherical harmonic coefficients for the incident illumination (Sec. 4.5). Finally, we refine the MVS geometry so that shading variations in the input images are properly explained by our image formation model and the estimated geometry (Sec. 4.6). The next sections review image formation using the SH illumination model and explain the illumination estimation and geometry refinement in detail. As we will describe, we handle concave surfaces and self-occlusion by computing the visibility of each vertex from all directions, and we adaptively tune the order of the SH approximation for higher accuracy in areas with higher ambient occlusion (i.e., more self-occlusion).

4.3 Image Formation Model

As introduced in Sec. 2.1.1, the reflection equation for Lambertian surfaces can be described by Eq. (2.2). In this chapter, we employ the same image formation model, but assume all surface points share the same albedo value, e.g. k_d. Then the albedo term can be absorbed by the lighting term, which is equivalent to scaling the lighting intensity by a constant factor. By denoting the scaled lighting term as $L_a(\boldsymbol{\omega}_i) = k_d L(\boldsymbol{\omega}_i)$, the reflection equation can be rewritten as

$$B(\boldsymbol{q}) = \int_{\Omega} L_a(\boldsymbol{\omega}_i) V(\boldsymbol{q}, \boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}, 0) d\boldsymbol{\omega}_i. \tag{4.1}$$

As discussed in Sec. 2.1.1, this equation can be simplified using SH. We will use different simplifying strategies respectively for lighting estimation and geometry refinement in the following sections.

4.4 Multi-view Stereo Reconstruction

We utilize a multi-view stereo method to achieve an initial reconstruction, which will then be used for lighting estimation and as an input geometry for shape refinement. As multi-view stereo has been researched for many years, any MVS method which is able to provide a coarse 3D reconstruction can be employed here. On the benchmark website[1], the method proposed by Furukawa & Ponce (2010) has achieved the highest accuracy on many evaluation data sets. This method firstly matches feature points across the different images to obtain a high-fidelity 3D reconstruction of these sparse points. Then, it expands this sparse reconstruction to a surface reconstruction based on photo-consistency constraints. Afterwards, this initial surface reconstruction is refined by minimizing the photo-consistency-based energy function once more. As can be seen from the algorithmic detail, this method works fairly well for textured objects. For the uniform-albedo objects that we are specifically focusing on, this method can also give a reasonable shape for surfaces with lots of geometric detail. However, for smooth surfaces on which no feature points can be found, this method usually fails to obtain the coarse structure. Based on this observation, we utilize this method for the data sets shown in Fig. 4.6 and Fig. 4.9.

Liu *et al.* (2010) propose another MVS method, which firstly reconstructs a point cloud by stereo matching and then fuses this point cloud to a watertight

[1] vision.middlebury.edu/mview/

mesh representation. In order to obtain better stereo matching results, it starts from the visual hull, which is obtained by space carving using the silhouette images. Besides, this MVS method has a robust scheme for detecting the depth points that were wrongly estimated, and it then replaces them with visual hull points. By doing so, this method is able to generate a robust reconstruction. For example, for smooth surface regions, it will not generate a worse shape than the visual hull, and thus will keep the coarse geometric structure in the reconstructed results. Although on the evaluation website, this method generates less accurate results, its robustness enables us to perform a robust lighting estimation and thus a successful shape refinement. This method is used for the data shown in Fig. 4.4, Fig. 4.7 and Fig. 4.10, where the smooth region is challenging to reconstruct using Furukawa's method Furukawa & Ponce (2010).

4.5 Lighting Estimation

With the initial model reconstructed using MVS, our method first estimates the SH coefficients for the incident illumination. Explicitly considering the visibility function in our image formation model enables us to reconstruct non-convex objects. Taking the same simplifying strategy as in Eq. (2.7), i.e. defining $T(\boldsymbol{q}, \boldsymbol{\omega}_i) = V(\boldsymbol{q}, \boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}(\boldsymbol{q}), 0)$, the image irradiance equation becomes

$$B(\boldsymbol{q}) = \sum_{l=0}^{N_D} \sum_{m=-l}^{l} L_{lm}^a \, T_{lm}(\boldsymbol{q}) = \sum_{k=1}^{n^2} l_k^a \, t_k(\boldsymbol{q}), \tag{4.2}$$

where N_D is the order of the SH, $n = N_D + 1$, and l_k^a and t_k are respectively the SH coefficients of lighting L_a and the visibility related term T. The irradiance B is known from the images, and the MVS geometry gives us an approximation for the visibility coefficient t_k. First, we use the model to compute the visibility of each vertex as a function of incident light direction. For each vertex, the coefficients t_k are the projection of the product of the visibility function and the clamped cosine function onto the SH basis functions. We calculate the coefficients $\boldsymbol{l} = \{l_1, \ldots l_{n^2}\}$ by minimizing the ℓ_1 norm of the difference between the measured and computed image irradiances at each mesh vertex:

$$\hat{\boldsymbol{l}} = \underset{\boldsymbol{l}}{\operatorname{argmin}} \sum_{i}^{N_v} \sum_{c \in Q(i)} \left| \sum_{k=1}^{n^2} l_k^a \, t_k(\boldsymbol{q}_i) - I_c(P_c(\boldsymbol{q}_i)) \right|. \tag{4.3}$$

Here, i is the vertex index, N_v is the number of vertices, c is the camera index, $Q(i)$ is the set of cameras that can see the i-th vertex $\boldsymbol{q}_i$, P_c is the projection matrix for camera c, and $I_c(P_c(\boldsymbol{q}_i))$ represents the image intensity corresponding to vertex i and captured by camera c. The ℓ_1 norm makes this estimation robust in the presence of outliers like interreflections, specularities, and errors in the MVS geometry.

We are estimating the low order SH coefficients for the illumination here. The specified order number is automatically decided by the local occlusion situation on the surface; see Sec. 4.7.

4.6 Shading-based Geometry Refinement

Given the current estimated geometry and illumination, the final step is to refine the geometry using shading information. For this step, we compute the visibility for each vertex using the current geometry, and assume that it does not change during the refinement. If we define $L_v(\boldsymbol{q}, \boldsymbol{\omega}_i) = L_a(\boldsymbol{\omega}_i)V(\boldsymbol{q}, \boldsymbol{\omega}_i)$, the image irradiance equation can be rewritten

$$B(\boldsymbol{q}) = \int_{\Omega} L_v(\boldsymbol{q}, \boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \boldsymbol{n}(\boldsymbol{q}), 0) d\boldsymbol{\omega}_i. \tag{4.4}$$

As discussed in Chapter 2, this is a convolution of L_v with the clamped cosine kernel determined by the surface normal. Similar to Eq. (2.5), according to the Funk-Hecke theorem Basri & Jacobs (2003), the convolution of two signals in the spatial domain results in the dot product of their SH coefficients in the frequency domain (SH domain here). Thus, the image irradiance equation can be expressed as

$$B(\boldsymbol{q}) = \sum_{l=0}^{N_D} \sum_{m=-l}^{l} \Lambda_l \, g_{lm} \, \hat{\rho}_{dl} \, Y_{lm}(\boldsymbol{n}(\boldsymbol{q})), \tag{4.5}$$

where g_{lm} is the SH coefficients of L_v, $\hat{\rho}_{dl}$ is the SH coefficients of the clamped cosine term and is known, and Y_{lm} is the SH function. The scalar Λ_l is defined as

$$\Lambda_l = \sqrt{\frac{4\pi}{2l+1}}. \tag{4.6}$$

Here, we have to allow the use of higher order spherical harmonic approximations when necessary (see Sec. 4.7). The function Y_{lm} depends only on the surface normal $\boldsymbol{n}$.

We run an optimization for each vertex position that attempts to minimize shading errors in all visible views. The computed irradiance is unlikely to match the observed irradiance, for many reasons: interreflections, radiometric calibration errors, approximation errors for the spherical harmonic illumination representation, and so on. Rather than directly comparing irradiance values, we compare the gradients of the observed and computed irradiances at each vertex. This is natural, because shading is expected to be more accurate for higher frequency shape components. Mathematically, we define the multi-view shading gradient error E_0 as

$$E_0 = \sum_{i}^{N_v} \sum_{j \in N(i)} \sum_{c \in Q(i,j)} \big(r_c(i,j) - s(i,j)\big)^2, \tag{4.7}$$

where i and j are vertex indices, $N(i)$ is the set of the neighbors of the i-th vertex, c is the camera index, $Q(i,j)$ is the set of cameras which see vertex i and j, and r and s are the measured image gradient and predicted shading gradient, respectively. We compute the gradients r and s with direct differences, namely,

$$\begin{aligned} r_c(i,j) &= I_c(P_c(\boldsymbol{q}_i)) - I_c(P_c(\boldsymbol{q}_j)), \quad \text{and} \\ s(i,j) &= B(\boldsymbol{q}_i) - B(\boldsymbol{q}_j). \end{aligned}$$

The shading value B is calculated according to Eq. (4.5). With the estimated illumination, the only remaining undefined variable in Eq. (4.5) is the normal $\boldsymbol{n}$, which we can compute from the vertices' positions. We limit vertex displacements to 3D locations that project into the object's silhouettes in all input views. Combining the silhouette and shading constraints gives the following new objective function E_1 for the multi-view shading gradient:

$$E_1 = \sum_{i}^{N_v} \sum_{j \in N(i)} \sum_{c \in Q(i,j)} d(i,j,c), \tag{4.8}$$

where i, j, $N(i)$, c, $Q(i,j)$ are the same as in Eq. (4.7). The function $d(i,j,c)$ has the following form:

$$d(i,j,c) = \begin{cases} (r_c(i,j) - s(i,j))^2, & M(\boldsymbol{q}_i) \cdot M(\boldsymbol{q}_j) \neq 0 \\ \infty, & \text{otherwise,} \end{cases} \tag{4.9}$$

where ∞ is a large constant that imposes a severe penalty if a vertex leaves the silhouettes, and M is a mask image which is non-zero inside the silhouettes and zero outside.

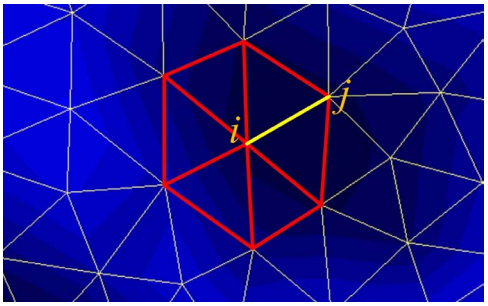


Figure 4.2: Anisotropic smoothness constraint: the smoothness weight for each edge is determined by the image gradient along the edge.

Smoothness constraint In practice, we have found that the shading gradient error alone leads to noisy reconstructions in areas where the normal is not sufficiently constrained or where errors in our image irradiance approximation are significant. Traditional smoothness terms might erroneously remove fine shape detail. We thus use an anisotropic smoothness constraint based on the image gradient that filters noise while preserving details captured by the shading gradient constraint.

We observe that for objects of uniform albedo, the image gradient can be used to infer geometric smoothness. We use a small smoothness weight in regions with large image gradients, allowing the shading constraint to capture fine detail. In areas where the image gradient is small, the shape is most likely smooth, so we use a larger smoothness weight. Fig. 4.2 shows this idea.

The smoothness constraint is imposed between vertex i and its 1-ring neighbors, with the weight being assigned to the corresponding edges. An isotropic smoothness constraint would require the geometric differences between vertex i and its neighbors to be as small as possible, with the same weight for each edge. Our anisotropic smoothness term, on the other hand, assigns different weights based on the image gradient between neighboring vertices. The weight of edge e_{ij}, for example, is determined by the corresponding image gradient in the camera most directly facing the vertex i. The weight for each edge is defined as

$$w_{ij}^{s} = 1 - \min(\hat{r}(i,j), C)/C, \tag{4.10}$$

where $\hat{r}(i,j)$ is the image gradient and C is a constant setting a lower bound on the smoothness weight when the gradient is large.

Combining the anisotropic weights with traditional mean curvature flow Meyer *et al.* (2002), the smoothness term E_2 becomes

$$E_2 = \sum_i \| \sum_{j \in N(i)} w_{ij}^s w_{ij}^m (\boldsymbol{q}_i - \boldsymbol{q}_j) \|_2^2, \tag{4.11}$$

where $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are the positions of vertex i and j, and w_{ij}^m is the common cotangent weight. The cotangent weight w_{ij}^m is defined as

$$w_{ij}^m = \frac{1}{2A_i}(\cot \alpha_{ij} + \cot \beta_{ij}), \tag{4.12}$$

where α_{ij} and β_{ij} are the two angles opposite to the edge $(\boldsymbol{q}_i, \boldsymbol{q}_j)$, and A_i is the Voronoi area of vertex $\boldsymbol{q}_i$.

We optimize a cost function summing the shading gradient E_1 and smoothness constraints E_2, defined as

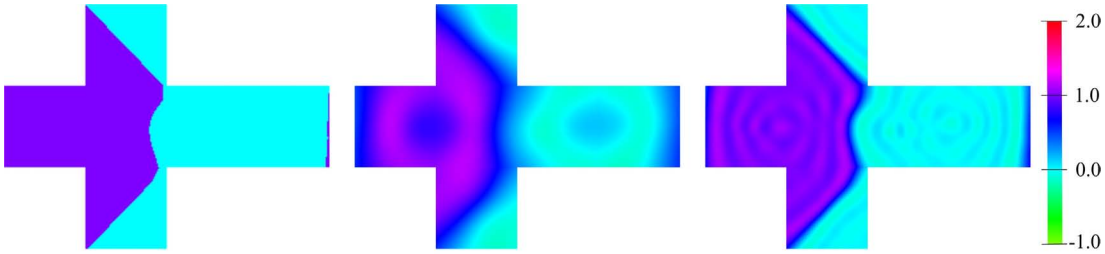
$$E = \lambda E_1 + (1 - \lambda) E_2, \tag{4.13}$$

where λ is a weighting factor. Optimizing all the vertex positions simultaneously is computationally intractable because of the non-linear SH function. Optimizing vertices one at a time, however, does not afford enough flexibility to adjust the local surface shape. Our algorithm visits each vertex in turn in a fixed order, optimizing the positions of a patch comprising the vertex and its 1-ring neighbors in each step. To avoid self-intersections as far as possible, we restrict vertex motion to be along the initial surface normal direction.

We could iterate by recomputing visibility using the refined geometric model, re-estimating lighting, refining the geometric model, and so on. In practice, however, we find that one pass suffices for an accurate reconstruction.

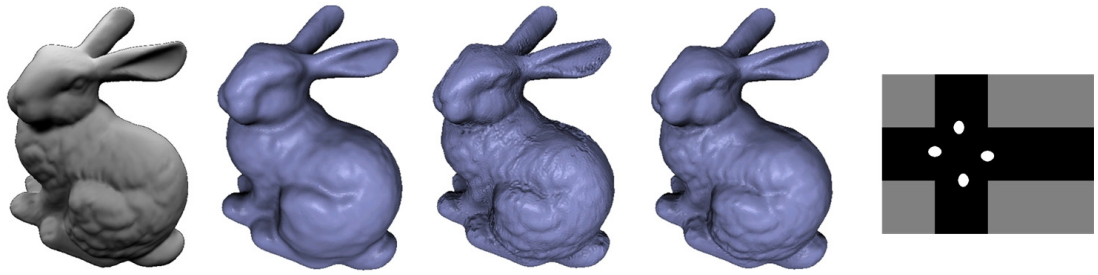
4.7 Adaptive Geometry Refinement

For convex Lambertian objects, low-order spherical harmonics suffice to approximate the irradiance well. For more complex objects, however, we must use high-order approximations, which are slower to compute. We use the local ambient occlusion Langer & Bülthoff (2000) to adapt the order of the SH approximation to the geometry. Ambient occlusion corresponds roughly to an integral over the local visibility hemisphere, so it is high for vertices with more local self-occlusion. We segment the mesh into two sets based on whether the ambient occlusion at each



(a) visibility map (b) representation by 4th order SH (c) representation by 14th order SH

Figure 4.3: An example visibility map and its SH representations of different order.



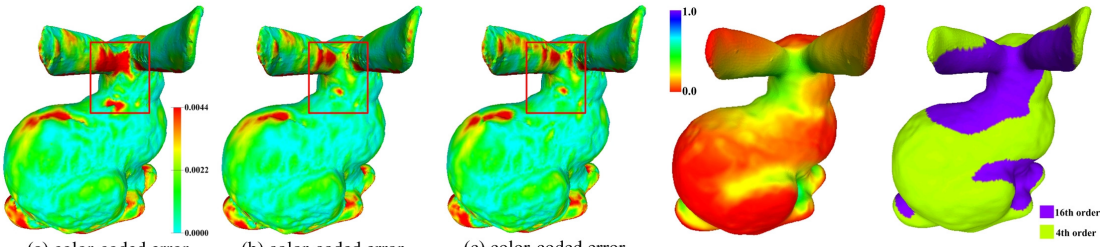
(a) image synthetically generated (b) MVS result (c) the reconstructed model without anisotropic smoothing (d) the final optimized model with anisotropic smoothing (e)environment map for synthetic data

Figure 4.4: Evaluation on synthetic data.

vertex is over a threshold, and use high-order and low-order SH approximations for vertices with high and low ambient occlusion, respectively (Fig. 4.5 (d)(e)). Although the SH approximation error depends on the specific visibility function at each vertex, not just its integral, we have found that the ambient occlusion gives a good balance between reconstruction accuracy and computational complexity.

4.8 Results

We validated our algorithm using a synthetic bunny model, shown in Fig. 4.4, and four real world data sets: an angel statue (Fig. 4.6), a sculpture of a fish (Fig. 4.7), a crumpled sheet of paper (Fig. 4.9), and a plaster cast of a face (Fig. 4.10). For the real world models, we took between 22 and 33 photos with a Canon $5D$ Mark II from calibrated positions. We captured images at the full camera resolution and cut out the region of interest containing the object, yielding images of around 800×600 pixels. For some models we also captured laser range



(a) color-coded error rendering for 4th order (b) color-coded error rendering for 16th order (c) color-coded error rendering for adaptive order (d) ambient occlusion (e) SH order adjustment

Figure 4.5: Adaptive geometry refinement: Our reconstruction using adaptive SH order (c) is almost as accurate as the high order case (b), which is obviously better than the low order case (a). The SH order used for every vertex (e) depends on its ambient occlusion value (d).

scans with a Minolta Vivid 910. We used Furukawa's method Furukawa & Ponce (2010) to generate the initial MVS models for the angel and the paper, and Liu's method Liu *et al.* (2010) for the bunny, the fish and the face. These MVS results are re-meshed to get a uniform triangulation, resulting in 30000 vertices for the bunny and 200000 vertices for the real scenes. We use DirectX to render a cube map for the visibility function at each vertex in the re-meshed result. Fig. 4.3 shows an example visibility map and its SH representations at different orders. For the synthetic model, we used 4 simulated area light sources (Fig. 4.4 (e)). The real objects were captured in two different environments: a large indoor atrium environment with a variety of light sources at different locations and distances (lighting I), and a room with several rows of standard office lighting on the ceiling (lighting II), Fig. 4.8. For the lighting estimation, we used conjugate gradient to solve the ℓ_1 minimization problem in Eq. (4.3). The shape is then refined by minimizing Eq. (4.13) using the Levenberg-Marquardt algorithm.

Parameters There are two tunable parameters in our method, λ in Eq. (4.13), and C in Eq. (4.10). Experimentally, we determined $\lambda = 0.3$ for all data sets. C was set to 20 for the bunny model, 100 for the angel model, and 50 for the other real-world models.

Generally, the selection of C depends on the level of image noise and uniformity of the albedos. For instance, less uniform albedos require a higher C. The per-vertex ambient occlusion threshold value (Sec. 4.7) was set to 0.1. 4-th order SH approximations were used for vertices with low ambient occlusion. Vertices

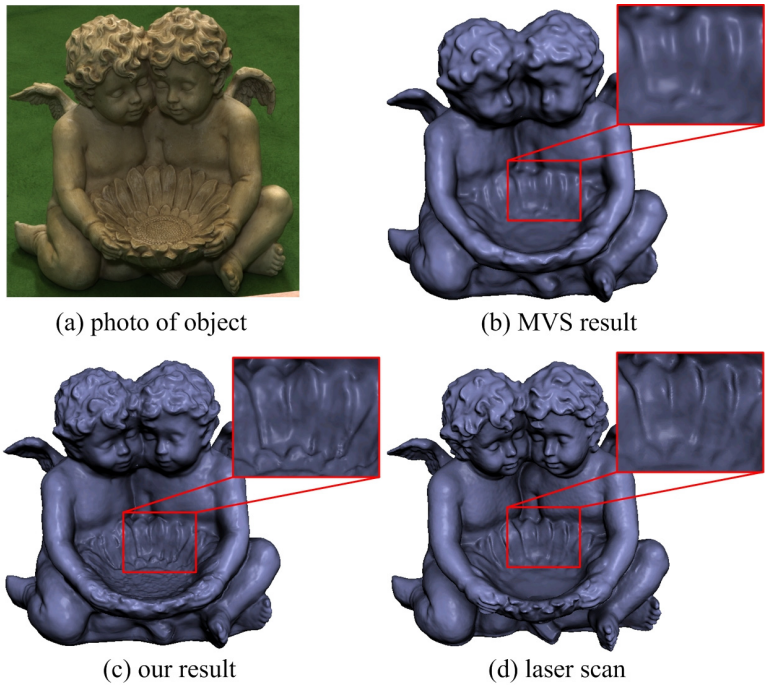
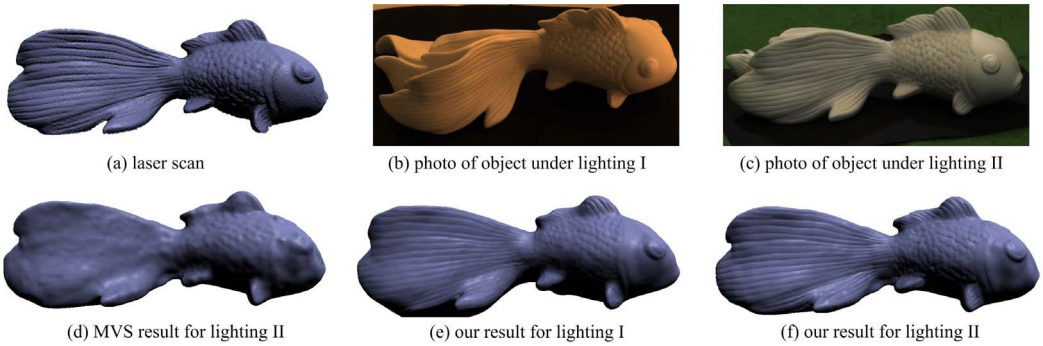


Figure 4.6: Our approach reconstructs models of much higher detail than state-of-the-art stereo approaches.

over the threshold used 14-th and 16-th order approximations for the real and synthetic sets, respectively.

Synthetic scene Our synthetic dataset was generated by rendering 20 images of the "bunny" model at 800×600 pixel resolution. Fig. 4.4 (a) shows an example image. The MVS result (Fig. 4.4 (b)) lacks fine-scale detail. Our refinement without anisotropic smoothing (Fig. 4.4 (c)) brings out more detail, but also has artifacts on the surface. In contrast, our complete reconstruction approach (Fig. 4.4 (d)) shows the high-frequency shape detail nicely with no disturbing artifacts. Table 4.1, a numerical evaluation of the reconstruction error w.r.t. the ground truth model, confirms the accuracy of our results.

Real-world scenes Our algorithm produces results of similarly high quality for the real objects shown in Figs. 4.6, 4.7, 4.9, and 4.10. While the MVS reconstruction consistently fails to capture high-frequency details, our algorithm produces results with an accuracy that rivals and sometimes exceeds the quality of a laser range scan. For instance, in Fig. 4.10 our approach not only brings out



(a) laser scan (b) photo of object under lighting I (c) photo of object under lighting II

(d) MVS result for lighting II (e) our result for lighting I (f) our result for lighting II

Figure 4.7: Fish reconstructed under two lighting conditions (cf. Fig. 4.8) — in both cases, our final result (e),(f) is much more detailed than the MVS results ((d) is only shown for lighting II and is better than MVS results for lighting I) and close to the laser scan (a).

	Position[‰]		Normal[deg.]		Runtime
	mean	std	mean	std	
MVS result	1.44	1.24	8.66	6.93	
adaptive, no smoothing	1.17	1.13	8.53	9.99	2 hours
adaptive + smoothing	1.15	1.07	7.05	**6.03**	2 hours
4th order + smoothing	1.19	1.13	7.28	6.28	**1 hour**
16th order + smoothing	**1.13**	**1.06**	**6.91**	6.17	4 hours

Table 4.1: Quantitative evaluation on synthetic data. First column: position error (in ‰ of bounding box dimension). Second row: error in surface normal direction in degrees. Third row: run time.

the birth marks and pimples in the skin, but also extracts ridges on the rubber cap that are completely masked by measurement noise in the laser scan. Although the angel statue in Fig. 4.6 has a slightly varying albedo, our algorithm achieves high-quality results. Thus, in practice the constant albedo assumption is not a strict limitation. Fig. 4.7 shows reconstructions of a fish figurine captured under two very different lighting conditions (lighting I and II). In both cases, our final model is very accurate and close to the laser scan.

Runtime performance The algorithm's run time depends on the mesh density, the SH order, and the cube map dimensions for rendering and SH projection. The bunny mesh has 30000 vertices and was computed using visibility cube maps with 64×64 facets. Using unoptimized code on a standard PC with a 2.66 GHz Core 2 Quad processor, rendering the visibility map takes 33 minutes, and optimizing the shape takes roughly 1 hour and 30 minutes. Higher SH orders improve

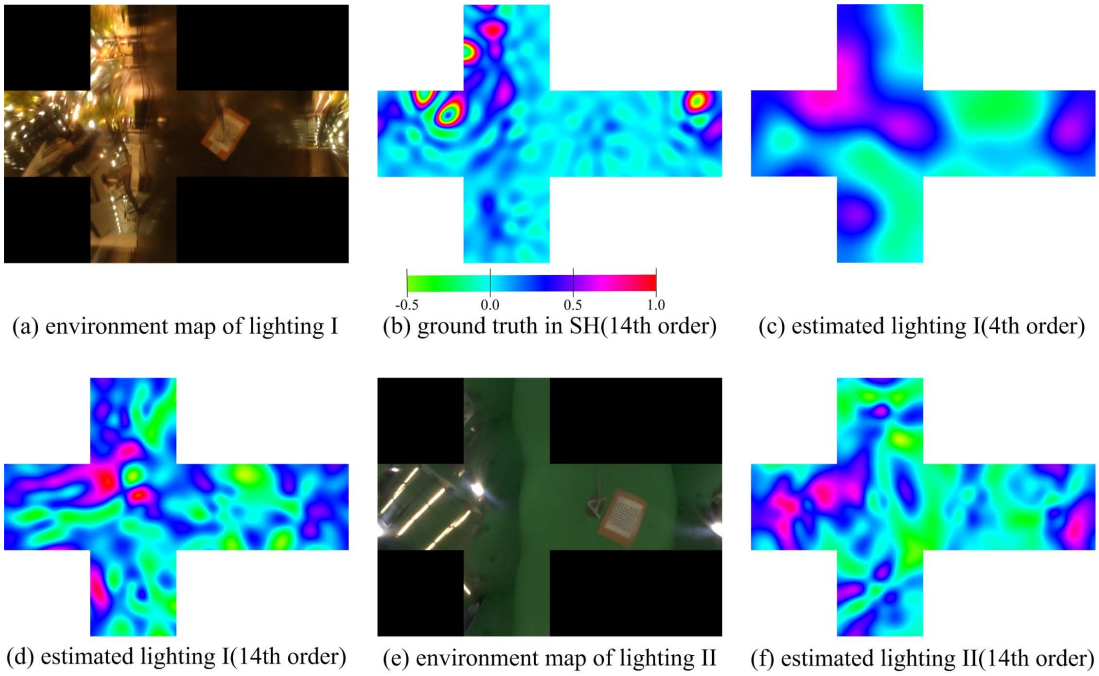


Figure 4.8: Comparing estimated lightings (c), (d), (f) to the captured environment map (a), (e) and the ground-truth SH representation of lighting I (b).

reconstruction quality, particularly in starkly occluded areas; see Fig. 4.5 (a), (b). Reconstruction of the bunny with 4-th order SH coefficients (Fig. 4.5 (a)) takes roughly 1 hour, but produces less accurate results than a full reconstruction with 16-th order (Fig. 4.5 (b)), which takes 4 hours to compute. Adaptive refinement reduces the runtime to only 2 hours with accuracy comparable to using high order coefficients throughout (Fig. 4.5 (c)).

Discussion The approach in this chapter is subject to a few limitations. The constant albedo assumption limits the possible application range. In the next chapter, we will modify the approach to handle clearly varying albedo. Another limitation comes from the assumption of Lambertian reflectance. This approach has difficulties when applied to non-Lambertian materials. We will also amend the approach to be applied to more general materials in later chapters. Also, this approach assumes a good initial guess of the geometry and would suffer from a failure of the MVS. In the future, we intend to start from a mesh obtained by active sensing methods Reynolds *et al.* (2011).

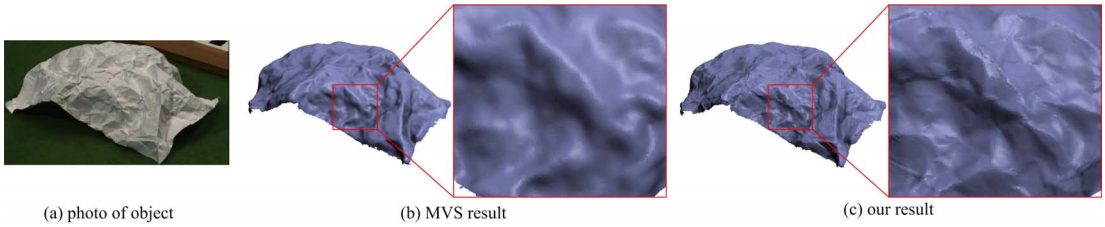


Figure 4.9: Our reconstruction of the crumpled paper recovers high frequency shape (c), while it is absent in the stereo result (b).

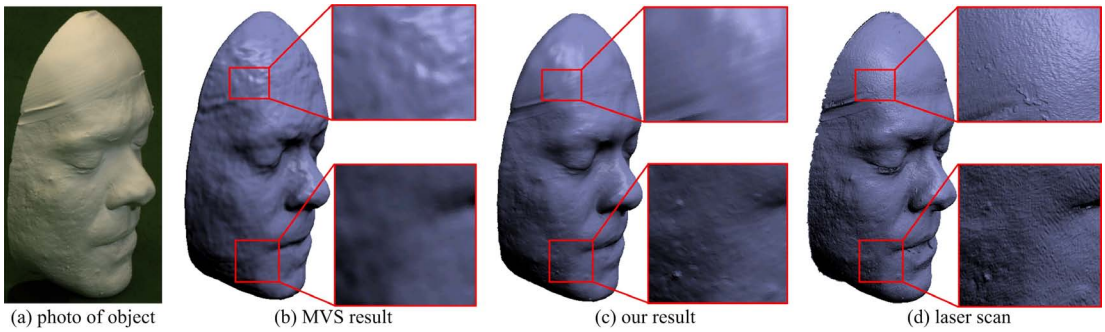


Figure 4.10: Reconstruction of a face plaster cast — the stereo result (b) lacks a lot of detail. Our reconstruction (c) captures even small-scale detail that, in the laser scan (d), is hidden by noise.

4.9 Conclusion

In this chapter, we demonstrate how to employ inverse rendering for scene reconstructions on static scenes. We proposed a new approach for purely image-based reconstruction of 3D models with extremely high surface detail. The core of the method is a shading-based refinement strategy for stereo reconstructions that succeeds under general unconstrained illumination. An efficient representation of visibility and lighting in the spherical harmonic domain enables the method to reliably estimate incident illumination and exploit it for high-quality shape improvement. Both visual and quantitative analysis show that our purely image-based results rival even laser range scans.

4.9 Conclusion

Part II

Dynamic Scene Reconstruction from Multi-view Video

As the idea of employing inverse rendering for scene reconstruction was demonstrated on static scenes in the previous chapter, in this part we are going to investigate the use of the inverse rendering concept for dynamic scene reconstruction from multi-view video input. Marker-less performance capture is one of the techniques able to produce temporal-coherent dynamic geometry from multi-view image sequences. However, previous methods are constrained by the capture environment, e.g. requiring controlled lighting or a green-screen background. One of the reasons is that they have not fully made use of or they have inappropriately modeled the information in the input video, especially the shading information in the images, which comes from the interaction between the lighting and the geometry of the scene.

With a better understanding of the light transport in the scene, in this part we exploit the shading information for dynamic scene reconstruction in two ways. Firstly, in Chapter 5 we look into the shading information in the video to add the true, fine geometric detail to coarse dynamic geometry to achieve high-quality performance capture under general and unknown illumination. Secondly, in Chapter 6, the shading cues are investigated for skeletal motion tracking under general unknown and even varying illumination, and a less-constrained background. With the techniques proposed in this part, we are able to achieve high-quality temporal-coherent dynamic geometry reconstruction, including the true fine-scale detail as well as accurate motion, under a less constrained environment, where the lighting could be general, unknown and varying.

Chapter 5

Shading-based Dynamic Shape Refinement under General Illumination

5.1 Introduction

Recent advances in computer vision and computer graphics have made it possible to reconstruct dynamic scenes from the real world into 3D mesh representations (e.g., Bradley *et al.* (2010); Cagniart *et al.* (2010b); de Aguiar *et al.* (2008); Vlasic *et al.* (2008)). This is achieved by capturing the scene from multiple synchronized video cameras and building the 3D shape from photometric cues, with the requirement that the reconstructions are geometrically and topologically consistent over time. These 3D shapes show plausible deformations up to medium scale detail, but often lack true detail at the finest level. As an example, a static laser-scan can be deformed to mimic the motion of the real scene, but any fine-scale detail thus obtained appears baked into the surface in the rest of the frames and does not capture the soft wrinkles on clothes and skin as can be observed from the images de Aguiar *et al.* (2008); Vlasic *et al.* (2008) (Fig. 5.1(d)). Some approaches attempt to reconstruct such detail through multi-view stereo from scratch or stereo-based refinement, but even then the detail in reconstructions is limited.

In this chapter, we propose a method that exploits knowledge about how a scene is lit and how it appears shaded in images to refine captured dynamic scene geometry. Shading information or photometric stereo cues have been exploited in certain previous approaches for capturing shape detail, for instance for facial per-

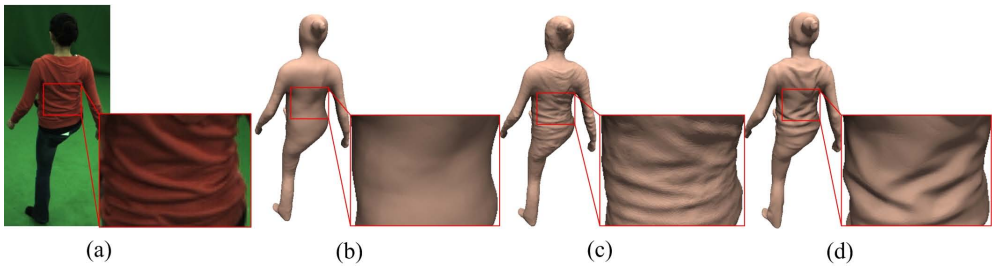


Figure 5.1: Shading based shape refinement: (a) captured image, (b) smooth model obtained by tracking, (c) our result of spatio-temporal shape refinement, (d) high-resolution geometry of a laser scan transferred by tracking, whose baked-in detail does not correspond to (a).

formance capture Wilson *et al.* (2010). However, they required controlled studio lighting through calibrated colored lights or a light stage, and made additional restrictive assumptions about the scene, such as that surface albedo is constant Hernandez *et al.* (2007). In contrast to these past methods, in this chapter, we propose a passive shape refinement method that attempts to reconstruct highly detailed spatio-temporally coherent 3D geometry under general illumination conditions (Fig. 5.1(c)).

We accept as input a sequence of multi-view images captured from a set of synchronized and calibrated cameras. Considering the state of the art in marker-less 3D motion capture systems (e.g., Gall *et al.* (2009)), we also assume that temporally coherent 3D meshes were reconstructed that lack any fine shape detail. We consider the estimated motion between the meshes to be accurate up to a coarse level. From this input, we try to capture high quality surface detail such as folds and deformation of the human body or cloth. For every time step of video, we explicitly estimate the incident illumination in the scene based on the reconstructed shape, make an estimate of the albedo distribution on the surface, and then use this information together with the lighting equation to recover the fine-grained structure and orientation of points on the surface. We assume a Lambertian model of reflection where incident lighting is given by an environment map that is parameterized in the spherical harmonic (SH) domain Ramamoorthi & Hanrahan (2001b), and where surface properties are given by a spatially-varying albedo map.

We mathematically formulate this in a maximum-a-posteriori (MAP) estimation framework, where we enforce a soft temporal coherency in estimated lighting, albedo and refined geometry. In this way, the environment map and surface

albedo can also change over time within reasonable bounds, e.g., when a subject walks across a room with several distributed lights, or when shifting apparel changes the albedo of a surface point over time. Our major contributions in this chapter, which were published in Wu *et al.* (2011a), are as follows.

1. We provide a method for adding spatio-temporally coherent millimeter scale surface geometry to coarse dynamic 3D scene models captured from multi-view video under general illumination.

2. We reconstruct time-varying incident illumination, time-varying and spatially varying surface albedo, and time-varying geometry detail, without using engineered lighting conditions.

3. We exploit the spatio-temporal information in the scene through soft temporal priors, which improves reconstruction quality but permits variations in the data.

5.2 Method Overview

We assume that a performance capture method was employed to obtain coarse mesh reconstructions, lacking true surface detail, at each time frame. We use the approach of Gall *et al.* (2009) that starts from a smoothed static model of the person of around 5000 vertices (this can be obtained through a static laser scan or imaged-based modeling) which it deforms to follow the motion in the scene. These spatio-temporally coherent meshes and the multi-view images captured under general unknown illumination form the input to our method. From this input, we perform spatio-temporal surface refinement at each frame to recover the high frequency geometry component by looking at shading cues. For refinement, we use a finer tessellated version of the coarse tracked geometry (vertex count increased to 80000), where a displacement for each vertex is found. In the rest of the chapter, we refer to the coarse estimates of vertex positions and normals given by the performance capture method as *low freq* and the refined vertex positions and normals output by our method as *high freq*. We perform this refinement successively at each frame to reconstruct the entire sequence.

Shading in the scene is generally an interaction result of lighting, material and geometry, which is described by the rendering equation Kajiya (1986) (see Sec. 2.1 for a detailed explanation). In the general reconstruction case, all these three components are unknown. To make the problem tractable, similar to Chapter 4,

5.2 Method Overview

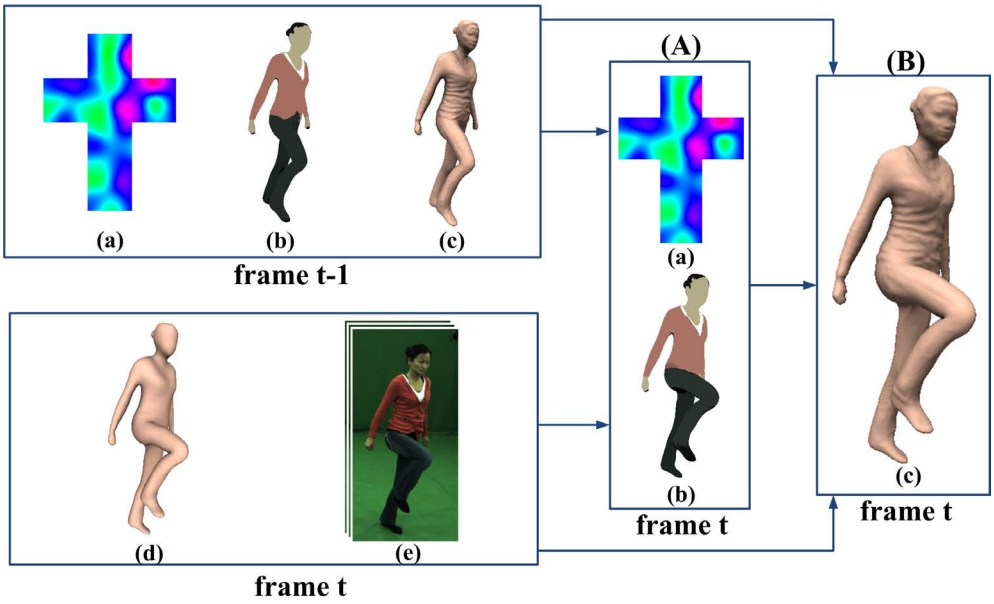


Figure 5.2: Overview — input to shape refinement at frame t: (a) lighting estimate at t-1, (b) surface albedo map at t-1, (c) detailed surface geometry at t-1, (d) coarse tracked model at t, (e) multi-view images at t. The two steps of our method: (A) lighting and albedo estimation, (B) recovery of high frequency shape detail.

we assume the surface to be Lambertian and employ spherical harmonics (SH) to represent the general lighting. So our refined model has three components: SH lighting coefficients, albedos and surface geometry (or positions of vertices $\{\boldsymbol{q}\}$). We formulate the problem of dynamic shape refinement as estimating these three components ($\{\boldsymbol{l}^t\}, \{\boldsymbol{k}_d^t\}, \{\boldsymbol{q}^t\}$) at a given frame using these estimates in the previous frame ($\{\boldsymbol{l}^{t-1}\}, \{\boldsymbol{k}_d^{t-1}\}, \{\boldsymbol{q}^{t-1}\}$) and the coarse performance capture model in the current frame ($\{\hat{\boldsymbol{q}}^t\}$). We develop a two-step algorithm that is visualized in Fig. 5.2. In the first step, we estimate the lighting coefficients and the surface albedos ($\{\boldsymbol{l}^t, \boldsymbol{k}_d^t\}$) at a given frame. These are estimated based on the lighting and albedos of the previous frame and the current tracked coarse model. In the second step, based on the estimated lighting and albedos, as well as the previous refined model, the high quality geometry at the current frame ($\{\boldsymbol{q}^t\}$) is recovered based on shading cues. We formulate these two steps as two MAP estimation problems with the appropriate priors, as detailed later in the following sections.

5.3 Image Formation Model

As introduced in Sec. 2.1.1, when the surface reflectance can be assumed to be Lambertian, we can simplify the reflection equation using SH to Eq. (2.5) or Eq. (2.7). We rewrite these two equations as follows:

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \sum_{l=0}^{N_D} \sum_{m=-l}^{l} L_{lm}\, T_{lm}(\boldsymbol{q}) = k_d(\boldsymbol{q}) \sum_{k=1}^{n^2} l_k\, t_k(\boldsymbol{q}), \tag{5.1}$$

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \sum_{l=0}^{N_D} \sum_{m=-l}^{l} \Lambda_l\, g_{lm}\, \hat{\rho}_{dl}\, Y_{lm}(\boldsymbol{n}), \tag{5.2}$$

where $B(\boldsymbol{q})$ is the reflected radiance, l_k is the SH coefficients for the lighting term, $t_k(\boldsymbol{q})$ are the SH coefficients for the combinational term of visibility and the clamped cosine function, $k_d(\boldsymbol{q})$ is the albedo value for surface point $\boldsymbol{q}$, N_D is the SH order employed and equals $n-1$, Λ_l is a constant scaling factor, g_{lm} are the SH coefficients for the visible lighting term, $\hat{\rho}_{dl}$ are the SH coefficients for the clamped cosine function, Y_{lm} are the SH basis functions, and $\boldsymbol{n}$ is the surface normal. We consider scenes captured using color images with RGB channels. The above equation, along with the equations derived in the following, hold true for all the color channels.

5.4 Lighting and Albedo Estimation

In the general case, the albedo varies across surface points. In an extreme case of high frequency texture with many surface albedos, solving for all the albedos and the incident illumination from the coarse geometry is infeasible. However, in most cases it is reasonable to assume that the albedo space is restricted and that the surface consists of patches of piecewise uniform albedo. For instance, most pieces of apparel have a dominant base color, as seen in Fig. 5.3. With a restricted albedo space we can simultaneously solve for albedo and lighting at each time step. Otherwise there would be an insufficient number of surface points (or shading samples) of similar reflectance seen under different orientations, which are needed to infer the incident illumination.

In our method, we first obtain an initial guess for the albedo of each vertex by making two assumptions: (i) that the lighting of the previous frame applies approximately to the current frame and (ii) an approximation to the *high freq*

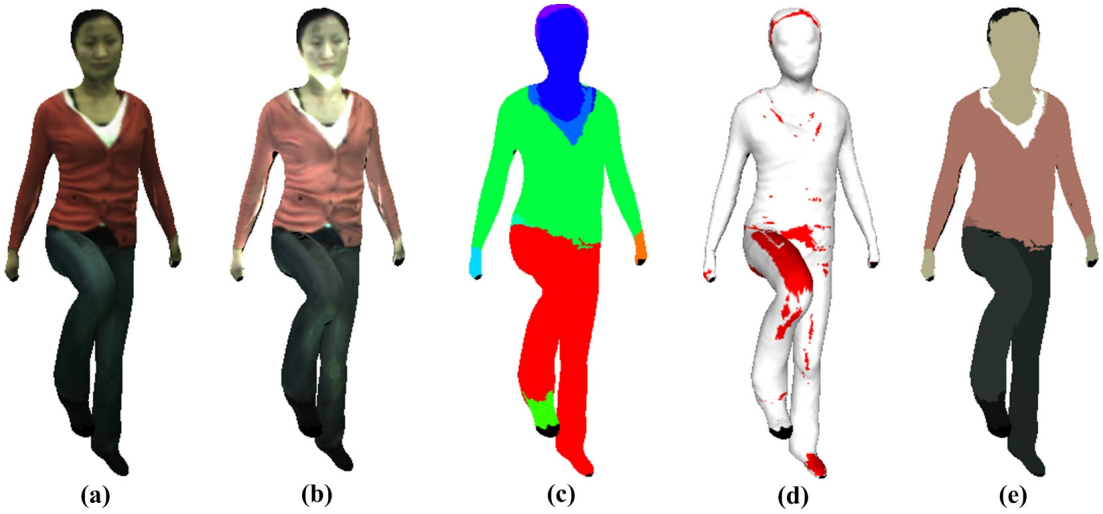


Figure 5.3: Stages for albedo estimation: (a) input textured model, (b) initial guess for albedos based on the previous frame's lighting, (c) albedo clusters detected on (b) through segmentation, (d) detected outliers marked in red, (e) the final albedo map.

surface normals at the current frame can be obtained by transferring the *high freq* normals of the previous frame through the *low freq* motion estimates given by the performance capture method (described in greater detail in Sec. 5.5). Using these initial guesses, we solve for the albedos at the current time step (i.e. an individual albedo for every mesh vertex) using Eq. (5.1) (Fig. 5.3(b)). Subsequently, we solve a global energy minimization problem to refine these albedo values over the entire shape, and to estimate the lighting conditions at the current frame.

Following our assumption about piecewise uniform albedo in the scene, we employ an image segmentation algorithm Felzenszwalb & Huttenlocher (2004) to segment the albedo map into surface parts of approximately constant albedo (see Fig. 5.3(c)). As criteria for segmentation, we provide the minimal size for each segment and the minimal difference in albedos across two segments (same parameters for all time steps).

Assuming we have K different albedo parts, we formulate a global problem that updates these albedo values as well as computes the lighting coefficients at the current frame. This is defined as a finding a MAP solution that maximizes the likelihood:

$$P(\boldsymbol{l}^t, \boldsymbol{k}_d^t | I^t) \propto P(I^t | \boldsymbol{l}^t, \boldsymbol{k}_d^t) P(\boldsymbol{l}^t) P(\boldsymbol{k}_d^t), \tag{5.3}$$

where $\boldsymbol{l}^t = \{l_1^t, \ldots l_{n^2}^t\}$ is the SH coefficients for the lighting, and $\boldsymbol{k}_d^t = \{k_d^t(1), \ldots k_d^t(K)\}$

represents the albedos for segmented parts. So the cost function we define is:

$$\psi(\boldsymbol{l}^t, \boldsymbol{k}_d^t) = \phi(I^t|\boldsymbol{l}^t, \boldsymbol{k}_d^t) + \phi(\boldsymbol{l}^t) + \phi(\boldsymbol{k}_d^t), \tag{5.4}$$

where $\phi(I^t|\boldsymbol{l}^t, \boldsymbol{k}_d^t)$ is the shading error, $\phi(\boldsymbol{l}^t)$ and $\phi(\boldsymbol{k}_d^t)$ are the priors for lighting and albedo in the current estimate. Specifically, as albedo segmentation may contain outliers, we use the ℓ_1 norm to define the shading error, i.e.

$$\phi(I^t|\boldsymbol{l}^t, \boldsymbol{k}_d^t) = |I(\boldsymbol{q}) - B(\boldsymbol{q})|\,. \tag{5.5}$$

We require the incoming light energy and the albedo of the surface points in the current frame to be not too different from those of the previous frame, which yields the priors:

$$\phi(\boldsymbol{l}^t) = \lambda_0(\sum_{k=1}^{n^2}(l_k^t)^2 - \sum_{k=1}^{n^2}(l_k^{t-1})^2)^2, \tag{5.6}$$

$$\phi(\boldsymbol{k}_d^t) = \lambda_1 \sum_{i=1}^{K}(k_d^t(i) - k_d^{t-1}(i))^2. \tag{5.7}$$

With the lighting and albedo estimated, we detect the outliers in the albedo segmentation for each part. Examples of outliers are surface points under cast shadows (where the first bounce illumination assumption is violated) or where the surface is non-Lambertian. To detect outliers, we calculate the median absolute deviation Rousseeuw & Leroy (1987) for each uniform albedo part as

$$\sigma_i = \alpha * median_{\boldsymbol{q}\in O(i)}\|I(\boldsymbol{q}) - B(\boldsymbol{q})\|_1, \tag{5.8}$$

where $O(i)$ represents the uniform-albedo part i and $\alpha = 1.4826$ is the theoretical correction factor Rousseeuw & Leroy (1987). If $\|I(\boldsymbol{q}) - B(\boldsymbol{q})\|_1 > \beta\sigma$, the surface point is considered as an outlier and will be optimized only by relying on the shape prior afterwards (in our experiments, we have set the penalizing threshold $\beta = 2.5$). We refine the lighting and albedo estimates again with these outliers excluded by solving Eq.(5.4) (Fig. 5.3(e)).

5.5 Recovery of High-frequency Shape Detail

Now, the lighting and the albedos for the current frame are known. The next step is to estimate the fine-scale geometry of the current frame based on the images, the coarse shape model at the current frame, and the refined model of

the previous frame. This can be defined as a MAP problem as well, the likelihood of which is:

$$P(\boldsymbol{g}^t|I^t, \boldsymbol{g}^{t-1}) \propto P(I^t|\boldsymbol{g}^t)P(\boldsymbol{g}^t|\boldsymbol{g}^{t-1}), \tag{5.9}$$

where $\boldsymbol{g}^t$ and $\boldsymbol{g}^{t-1}$ are the geometry of the current frame and the previous frame, and I^t are the current captured images. The cost function to optimize is thus

$$\psi(\boldsymbol{g}) = \phi(I^t|\boldsymbol{g}^t) + \phi(\boldsymbol{g}^t|\boldsymbol{g}^{t-1}), \tag{5.10}$$

where $\phi(I^t|\boldsymbol{g}^t)$ is the shading error and $\phi(\boldsymbol{g}^t|\boldsymbol{g}^{t-1})$ is the prior for the current geometry based on the previous frame's geometry.

The shading error measures the difference between the observed and predicted irradiances at each vertex according to the shape estimate. We are not comparing irradiances, since that comparison is less robust if the assumptions on lighting and image-formation are not exactly met. When evaluating the energy, we use grayscale intensities, instead of treating the three color channels separately. Our shading error is defined as:

$$\phi(I^t|\boldsymbol{g}^t) = \sum_{i}^{N_v} \sum_{j \in N(i)} \sum_{c \in Q(i,j)} |r_c(i,j) - s(i,j)|, \tag{5.11}$$

where i and j are vertex indices, N_v is the number of vertices, $N(i)$ is the set of the neighbors of the i-th vertex, c is the camera index, $Q(i,j)$ is the set of cameras which see vertex i and j, and $r(i,j)$ and $s(i,j)$ are the measured image gradient and predicted shading gradient, respectively.

An important step to solving this equation is determining $Q(i,j)$, which depends on the current estimate of the 3D geometry (vertex positions $\boldsymbol{q}_i$). A discrepancy between the hypothesized scene geometry and the real geometry will lead to wrong assumptions about what surface point is visible from what camera. Such errors translate into wrongly evaluated shading cues, and thus geometry artifacts. Fig. 5.4 shows one such error that often arises around a visibility shadow that more frontal geometry casts onto more distant geometry.

In Chapter 4, we have proposed a similar shading error metric to Eq. (5.11) for the reconstruction of static 3D scenes. However, there we assume a much denser set of input camera views (> 20) and better initial geometry to start with. In contrast, performance capture methods Gall *et al.* (2009) typically use only 8-12 cameras, and reconstruct a geometry that is only accurate up to a coarse scale. This makes the errors in determining $Q(i,j)$ more damaging for our situation, and demands explicit consideration. In order to implicitly downweight

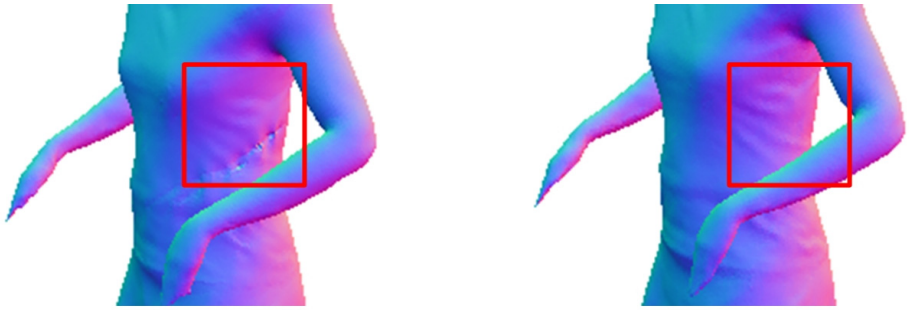


Figure 5.4: Handling errors in estimated geometry: (a) the geometry of the forearm is not estimated fully correctly — using an ℓ_2-metric shading term, this yields artifacts around the visibility shadow on the torso; (b) the ℓ_1-metric shading term prevents this artifact.

the influence of these errors, without having to resort to more complex visibility computation, we employ the robust ℓ_1 metric in Eq. (5.11) in contrast to the ℓ_2 metric (Fig. 5.4).

The prior $\phi(\boldsymbol{g}^t|\boldsymbol{g}^{t-1})$ enforces weak temporal coherence by requiring the current *high freq* normal field not to be much different from the one in the previous time step transformed into the current time step:

$$\phi(\boldsymbol{g}^t|\boldsymbol{g}^{t-1}) = \lambda_2 \sum_i \sum_{u,w} [\hat{\boldsymbol{n}}_i^t \cdot (\boldsymbol{q}_u^t - \boldsymbol{q}_w^t)]^2, \tag{5.12}$$

where $\boldsymbol{q}_u^t$ and $\boldsymbol{q}_w^t$ are the positions of vertices u and w, vertices u, w and i belong to the same mesh triangle, and $\hat{\boldsymbol{n}}_i^t$ is the propagated surface normal at vertex i based on the already reconstructed *high freq* normal field of the previous frame. This propagation is done by estimating the relative transformation R_i of the *low freq* normals between the two frames, using a method similar to Nehab *et al.* (2005), such that:

$$\tilde{\boldsymbol{n}}_i^t = R_i \tilde{\boldsymbol{n}}_i^{t-1}, \tag{5.13}$$

where $\tilde{\boldsymbol{n}}_i^t$ and $\tilde{\boldsymbol{n}}_i^{t-1}$ are the *low freq* normals of the current frame and the previous frame, respectively. Then we obtain the propagated fine-scale normal of the current frame by transforming the *high freq* normal of the previous frame as:

$$\hat{\boldsymbol{n}}_i^t = R_i \boldsymbol{n}_i^{t-1}, \tag{5.14}$$

where $\boldsymbol{n}_i^{t-1}$ is the normal of the refined model of the previous frame. We now obtain an initialization for the fine geometry at the current frame by displacing

(a) (b) (c)

Figure 5.5: Importance of temporal shape prior: (a) captured image, (b) reconstructed model using no temporal shape prior (*OneFrm* method), (c) improved reconstructed model using temporal shape prior.

vertex positions so as to align with the propagated normal field $\{\hat{\boldsymbol{n}}_i^t\}$. Starting from this initial estimate, the final refined vertex positions (and normals) are found by optimizing Eq. (5.10).

In our shape refinement procedure, we give the shading term less influence when optimizing regions with low albedo. This is because such regions suffer more from camera noise. We thus include a weighing term λ_2 in the shape prior Eq. (5.12):

$$\lambda_2 = \beta_1(2 - k_d(u)/\max_i(k_d(i))), \tag{5.15}$$

where $k_d(u)$ is the albedo for the vertex u, which is to be optimized, and $\max_i(k_d(i))$ is the maximum albedo of the current model.

Since optimizing the positions of all the vertices simultaneously might take too long, similar to Chapter 4, we adopt a patch-based optimization strategy that divides the surface into a set of patches and optimizes on the set of vertices belonging to each patch sequentially. This arrives at a local optimum that is usually quite robust.

5.6 First Frame Reconstruction

For the first time step, we cannot employ our spatio-temporal reconstruction scheme as information from the prior time instant is not available. Instead, we employ a static refinement method (referred to as *OneFrm*) that only uses image and coarse model information for the one time step under consideration. To this end, we first segment the shape into parts of uniform color, and assume that

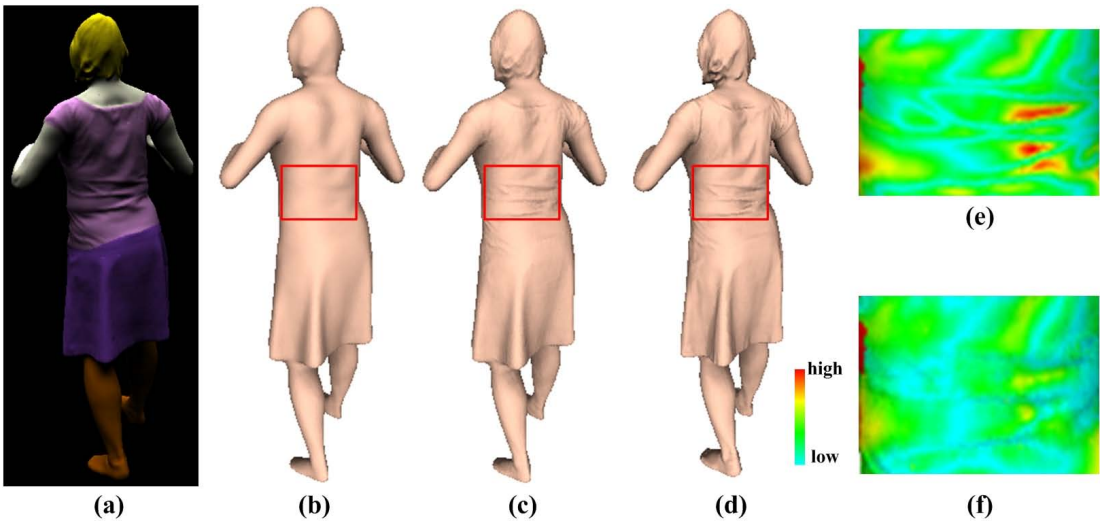


Figure 5.6: Shape refinement results on synthetic data: (a) one of the rendered images that we provided as input, (b) the smooth *low-freq* model obtained by tracking, (c) our spatio-temporal shape refinement result, (d) ground-truth model, (e) difference from ground truth for (b) in the inset region (color-coded error w.r.t. ground truth — red=high), (f) difference from ground truth for (c).

these are regions of uniform albedo. We then estimate lighting coefficients and albedo values using our method of Sec. 5.4, but without temporal priors. Next, we recover the high frequency surface detail using a spatial smoothness prior (used in Chapter 4) in Eq. (5.10) (instead of the shape prior from the previous frame) that requires neighboring surface vertices in a one-ring to have similar positions. This gives a reasonable estimate for the first frame. In later frames, however, we always resort to the full spatio-temporal scheme which is clearly better than using the static scheme sequentially to all time steps (see Fig. 5.5 and Sec. 5.7).

5.7 Experiments

We test our algorithm on one synthetic sequence for quantitative evaluation, and four real-world sequences for qualitative validation. We use the performance capture method of Gall et al. Gall *et al.* (2009) that uses an initial smooth mesh of around 5000 vertices to track the performance. We obtained this by smoothing a static laser scan of the performer (for real data, e.g., Fig. 5.1(b)) or by smoothing a ground truth input mesh (for synthetic data, Fig. 5.6(b)). Refinement is computed on the 80000 vertex versions of the coarse models (see Sec. 5.2).

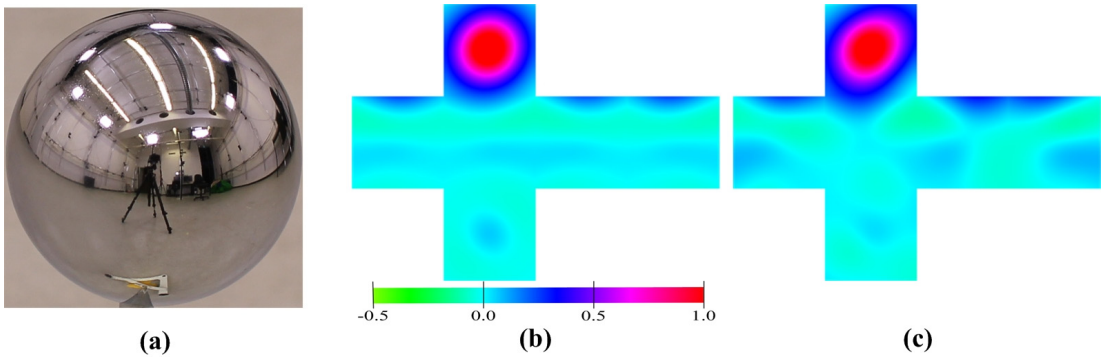


Figure 5.7: Lighting estimation: (a) typical light-source distribution for real-world datasets we used, (b) ground-truth lighting for synthetic data — SH approximation of the incoming radiance displayed onto a cubemap, (c) the estimated lighting for synthetic data by our algorithm.

Synthetic Scene We rendered a synthetic motion sequence of a female dancer of length 60 frames from 12 circularly arranged virtual cameras of resolution 1296×972. Surface albedo distribution was manually specified (5 regions of similar albedo), and the scene was rendered using a single area light from an overhead position (Fig. 5.6(a) shows one rendered frame).

We applied the performance capture method to all the frames; we then performed static refinement (*OneFrm*) on the first time step and spatio-temporal refinement on all subsequent ones. Fig. 5.6(c) and Fig. 5.6(d) show the refined model and the ground truth, respectively.

We compare the accuracy w.r.t. ground truth of the lighting and albedo estimation between the *OneFrm* and spatio-temporal refinement methods in Fig. 5.8(a) and (b). We use the normalized correlation coefficient to compare the estimates. This figure clearly demonstrates that by using spatio-temporal information for estimating lighting and albedo values, higher accuracy is achieved. Fig. 5.7 shows a visual comparison of our lighting estimate with the ground truth on the synthetic data, illustrating the high quality of our estimate.

We also evaluated the accuracy of the reconstructed high-resolution geometry from our algorithm. In Fig. 5.8(c,d,e,f), we show the errors in normal orientation and position as compared to the ground truth. Here, we also compare our method to the *OneFrm* method, and to the coarse tracked model as the baseline. These figures illustrate that our method reliably captures high-frequency shape detail that is not present in the coarse model. The refinement with *OneFrm* is understandably less accurate, especially in estimating normal orientations. The same

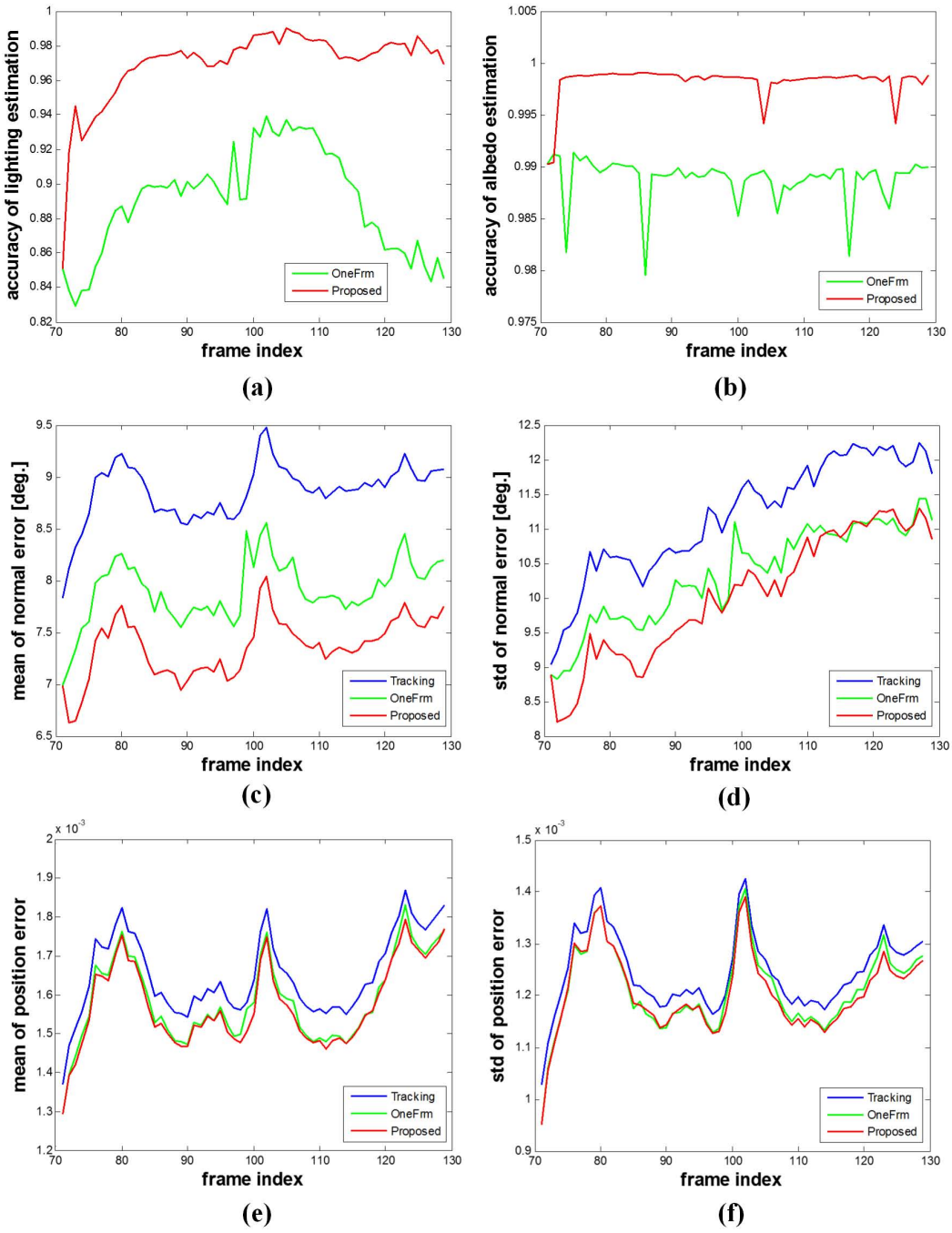


Figure 5.8: Quantitative evaluation on synthetic data: (a) lighting estimation accuracy, (b) surface albedo estimate accuracy, (c) mean of the estimated normal error, (d) std of the estimated normal error, (e) mean of the position error (normalized using the diameter of the bounding sphere of the model), (f) std of the position error. Our spatio-temporal shape refinement (red curve) yields the best results.

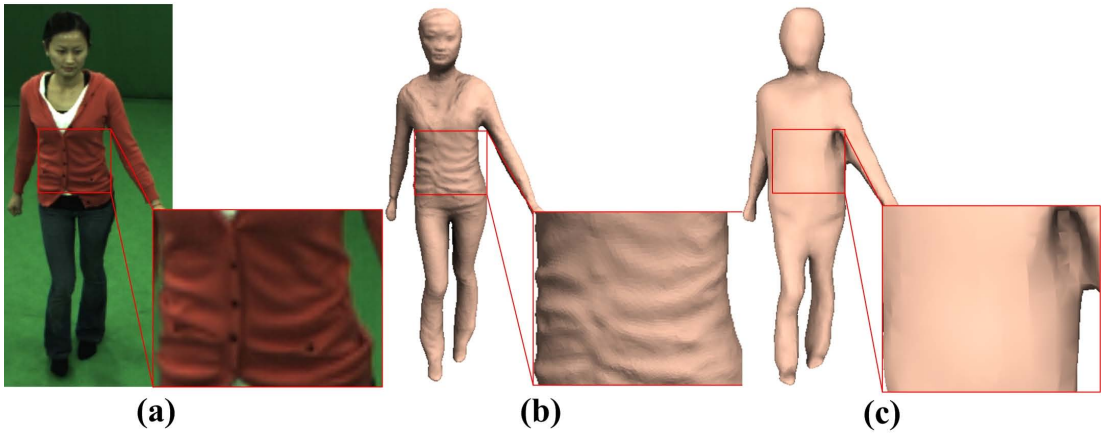


Figure 5.9: Qualitative comparison with stereo refinement: (a) captured image, (b) our shape refinement results, (c) stereo-based shape refinement of Liu et al Liu *et al.* (2010)

can be visualized in Fig. 5.5(b,c) as spatio-temporal refinement better brings out high frequency shape details. Using the *OneFrm* method independently at each frame also produces temporal flicker which is absent in the reconstructions of our method.

Real-world Scenes We validate our algorithm on four real captured sequences, showing 3 different subjects in different types of apparel. All sequences were captured indoors with non-engineered lighting, i.e. several area light sources and spot lights on the ceiling (Fig. 5.7-a). The results of shape refinement on certain frames are provided here. The first two sequences show an actress wearing a sweater and jeans performing different motions, namely walking (Fig. 5.9) and kicking (Fig. 5.1), the third shows another actress in a skirt performing samba dancing (Fig. 5.10-a,b), and the fourth shows an actor executing a Capoeira move (Fig. 5.10-c,d). For the first two sequences, 12 cameras at a resolution of 1296×972 pixels are used to record at a frame rate of 44 fps. For the latter two sequences that were provided to us by the authors of de Aguiar *et al.* (2008), 8 cameras running at the resolution of 1004×1004 pixels are used. We show the results in Fig. 5.10. In all cases, our method recovers the true dynamic detail seen in the images reliably. Our reconstructions capture the true time-varying detail visible in input images, as opposed to the deforming embossed static shape detail seen from performance capture methods that deform a (unsmoothed) static laser scan (Fig. 5.1(d)). In Fig. 5.9(c), we show a qualitative comparison of our

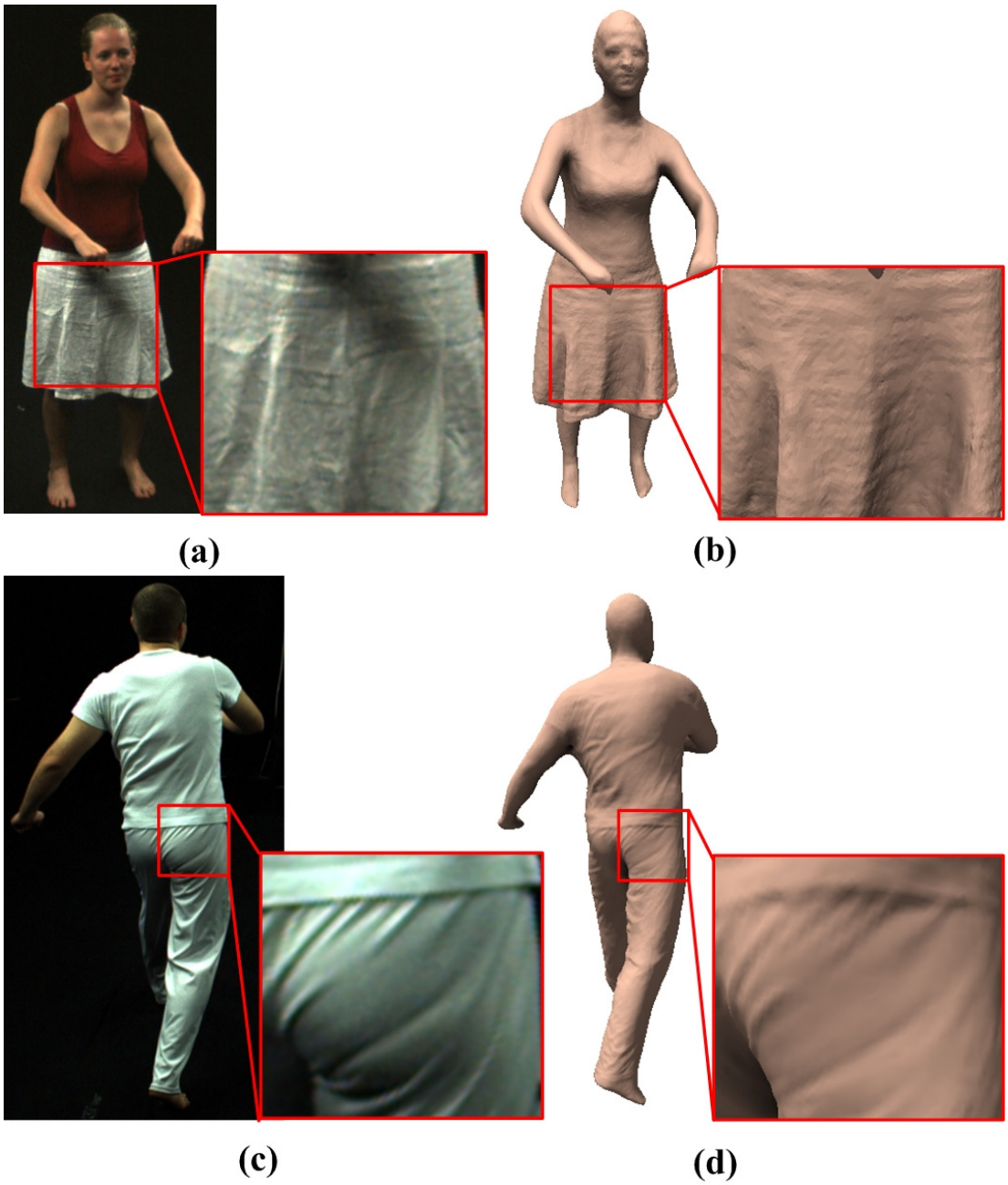


Figure 5.10: Qualitative evaluation on real datasets: (a,c) captured image, (b,d) our shape refinement results. Our refinement method faithfully brings out fine-scale detail from images.

method with a stereo based reconstruction method of Liu *et al.* (2010). It can be observed that our method brings out finer detail than stereo.

Runtime Performance We measured the runtimes of the various algorithmic components on a standard PC with a 2.66 GHz Core 2 Quad processor. Performance capture using Gall *et al.* (2009) takes on average $5-10$ s per time step. Per vertex visibility computation (one visibility environment map per vertex) takes around 10 minutes per frame. The shape refinement step takes around 6 minutes per frame. Since these three steps can be executed in parallel for processing sequences, the runtime is decided by the visibility computation step (10 minutes per frame).

Discussion On parts of the shape where image resolution is limited (for example, on the faces of the actors), our approach cannot completely recover the fine-scale detail. Reconstruction quality depends on the tracking accuracy of the performance capture approach; large tracking errors or tracking failure will lead to incorrect shape refinements. Thus, although the shape refinement potentially works for changing illumination and general background, the motion tracking step limits the whole pipeline so it only succeeds under constant lighting and with a green-screen background. In next chapter, we will show how to address this issue by also exploiting the shading information for motion tracking. Another limitation is that we assume Lambertian surfaces; as such, our algorithm fails to obtain the high-frequency detail on non-Lambertian parts of the shape. We will also address this issue in later chapters. Also, the assumption that the surface can be clustered into regions of uniform albedo is restricting and can be violated in some scenes. If too many different materials are present, the space of shading samples may not be sufficient in order to estimate albedo and lighting at the same time. In such cases, they may have to be spatio-temporally solved over more time instants, which makes the approach more vulnerable to tracking errors. In the future, we intend to handle this case by incorporating more priors on lighting and albedos.

5.8 Conclusion

In this chapter, we investigated the idea of employing inverse rendering for dynamic reconstruction of high-frequency shape detail. Specifically, we proposed

a general method for capturing high-quality time-varying surface detail by analyzing the shading information of multi-view video sequences captured under general illumination. We make minimal assumptions about the nature of the scene, the type of motion or the lighting requirements. Starting off from coarse per time-step reconstructions, we recover incident illumination, surface albedo and fine-scale surface detail in a spatio-temporally coherent way. Our reconstruction framework uses weak temporal priors to boost reconstruction quality, and it is able to allow for and capture temporal variations in lighting, albedo and shape.

5.8 Conclusion

Chapter 6

Full Body Performance Capture under Varying and Uncontrolled Illumination

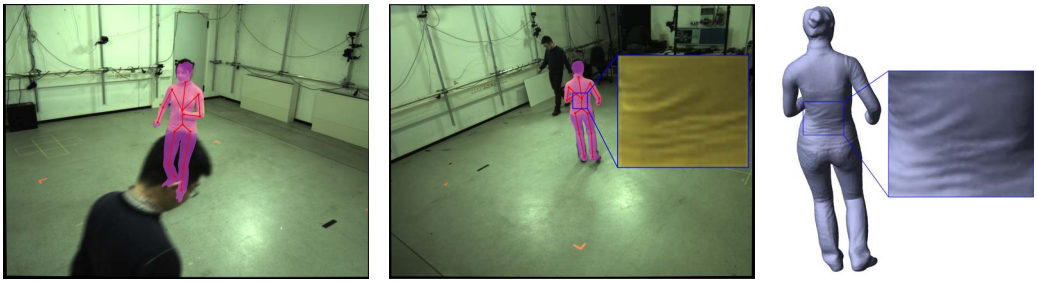
6.1 Introduction

In Chapter 5, we address the problem of how to reconstruct dynamic shape detail under general and unknown lighting. While the high-frequency geometry can be estimated by utilizing the shading information without resorting to specific lighting conditions, the low-frequency geometry, which has so far been reconstructed with a traditional template-based performance capture method Gall *et al.* (2009), still requires constant and well-controlled lighting and a green-screen background. This constrains this method to controlled indoor studio setups. In this chapter, we develop a new performance capture method which works under general, unknown and time-varying illumination, and a less-constrained background. The key step to achieve this goal is to also exploit inverse rendering for human skeletal motion tracking under general and varying illumination. The low-frequency geometry obtained by this step is further refined based on the shading cues to achieve a high-quality dynamic scene reconstruction. In conjunction, this combination of exploiting inverse rendering for pose estimation and shape refinement enables us to greatly broaden the application range of marker-less performance capture.

Marker-less capture of human skeletal motion from images is one of the well-studied problems of computer vision, with recent advances being able to reconstruct human motion at increasing speed and accuracy and under less-controlled

situations Balan *et al.* (2007); Bregler *et al.* (2004); Deutscher *et al.* (2000); Poppe (2007); Sidenbladh *et al.* (2000); Sigal *et al.* (2010); Stoll *et al.* (2011). These methods have several applications in industry, ranging from game and movie productions to use in biomechanics, ergonomics and sports sciences. However, despite great algorithmic advances, even the latest approaches cannot yet be applied in arbitrary environments with possibly changing lighting conditions, occlusions and starkly varying scene backgrounds. This is why purposefully placing markers in the scene is still the method of choice under such challenging conditions Raskar *et al.* (2007). Special effects professionals and producers of 3D video content are sometimes interested in factors beyond kinematic motion parameters — demanding faithful and detailed dynamic 3D shape models of captured scenes, such that believable virtual actors or convincing novel viewpoint renderings can be created. To respond to this requirement, performance capture methods Cagniart *et al.* (2010a); de Aguiar *et al.* (2008); Starck & Hilton (2007); Vlasic *et al.* (2008) are developed to simultaneously capture shape, motion and possibly appearance of people in general apparel from a handful of video recordings. Unfortunately, these methods are similarly limited to studio settings with controlled lighting, controlled background, and to scenes without static or dynamic occluders. This has prevented the use of performance capture in practical applications such as outdoor movie sets or sports stadiums.

In this chapter, we make a principal contribution towards the goal of model-based performance capture under less controlled conditions. We propose an algorithm that analyzes shading information to simultaneously estimate (a) human skeletal motion parameters, (b) arbitrary and time-varying incident scene illumination, (c) an approximation of surface reflectance, and (d) detailed dynamic shape geometry — such as folds and muscle bulges. We accept as input a multi-view video recorded from a synchronized and calibrated set of cameras, along with a rough initial shape-template of the person given as a 3D mesh fit to a kinematic skeleton. We do not require the subject to wear specific clothing or markers. Unlike previous performance capture methods Cagniart *et al.* (2010a); de Aguiar *et al.* (2008); Gall *et al.* (2009); Starck & Hilton (2007); Vlasic *et al.* (2008), we do not require a fully controlled scene background, such as a green screen, and thus do not expect exact foreground-background segmentations. We handle changing backgrounds and even some occlusions in the scene (Fig. 6.1). We do not rely on image features such as SIFT; our method is suitable even when the subject wears sparsely textured clothing.



(a) (b) (c)

Figure 6.1: Shading based pose tracking: (a,b) overlay of estimated pose with recorded images — the actor is partially occluded by a person moving in the background, (c) reconstructed high-detail 3D geometry. The inset shows folds of the yellow T-shirt captured in 3D.

The main idea of employing inverse rendering for skeletal motion tracking is to mathematically formulate the image shading constraint in terms of its differential w.r.t. the motion parameters of the kinematic chain representing the human body pose. Along with pose, we simultaneously estimate time-varying incident illumination, surface albedo and detailed surface geometry in a joint framework. Thus, we integrate the human motion estimation problem into the broader framework of multi-view shape-from-shading.

To summarize, the major contributions described in this chapter, published in Wu *et al.* (2012), are as follows.

1. We present a new theoretical formulation of performance capture that simultaneously recovers human articulated motion and time-varying incident illumination, by minimization of a shading-based error.

2. We provide a solution to reconstruct both skeletal motion estimates and finely detailed time-varying 3D surface geometry for human performances that are recorded under general and changing illumination, and in front of less constrained backgrounds.

6.2 Method Overview

The input to our method is a multi-view video sequence of a moving actor captured using a sparse set of synchronized and calibrated cameras. Lighting in the

6.2 Method Overview

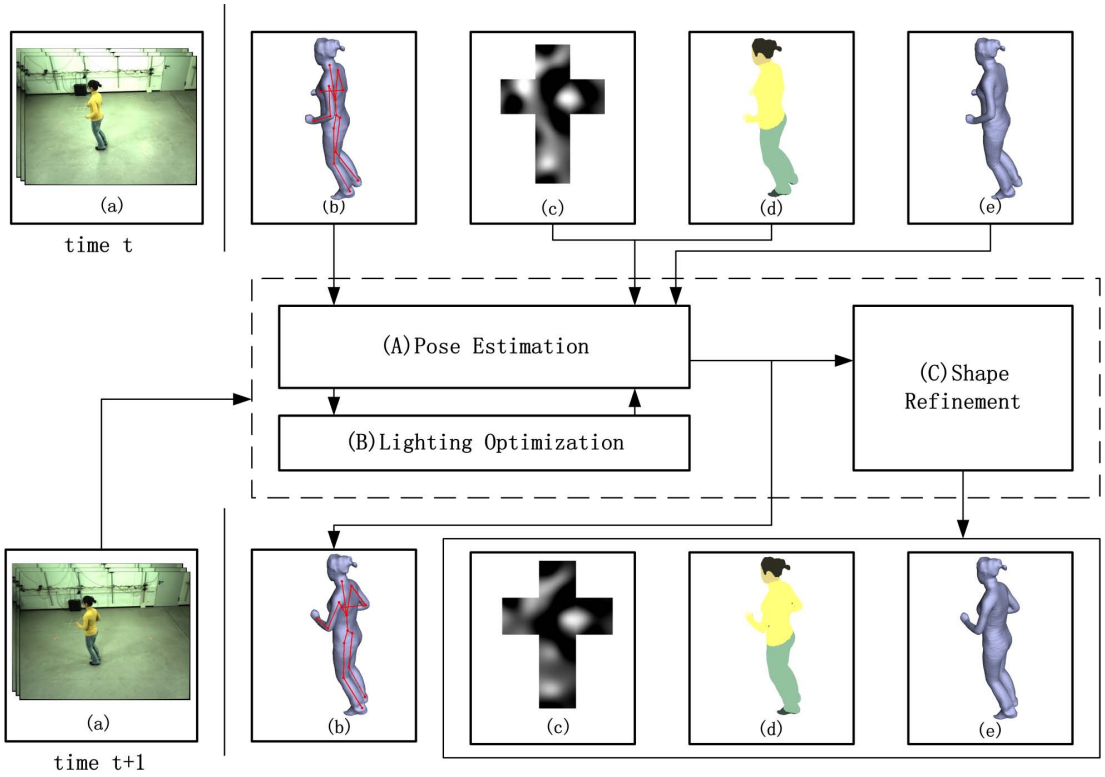


Figure 6.2: Overview: (a) input multi-view images, (b) skeletal pose, (c) incident illumination, (d) surface albedo, (e) refined surface geometry. (b-e) are outputs of our method. Steps (A,B) for estimating pose and lighting are alternated in a joint optimization framework. In step (C), final estimates of lighting, albedo and surface geometry are obtained. These estimates at t are provided as input for the optimization at $t + 1$.

scene can be arbitrary and time-varying, and since no background subtraction is required, no green-screen is expected, and other potentially occluding elements can be in the scene. A rigged 3D mesh model with an embedded skeleton is provided as a template for tracking. We only need a smooth template mesh at a low resolution; the fine-scale detail is added later by our method. Similar to Gall *et al.* (2009), the smooth template is built from a static laser scan of a person. Alternatively, image-based reconstruction methods are also feasible to reconstruct a template model directly from images. The embedded skeleton, as well as the skinning weights for each vertex (which connect the mesh to the skeleton) are obtained using standard tools.

An outline of the processing pipeline is given in Fig. 6.2. Given a set of cap-

tured multi-view images (a) as input, at each time-step $t+1$ we estimate skeletal pose (b), incident illumination (c), surface albedo (d), and detailed surface geometry (e). For each of these variables, we solve an inverse-rendering problem that attempts to bring the rendered images as close as possible to the captured image data. In Step A, starting with the skeleton and the refined mesh from time t, the skeletal pose is optimized by assuming incident lighting and surface albedo from t, thereby exploiting temporal coherence. In Step B, the incident illumination at time $t+1$ is estimated based on the skinned coarse mesh in the new skeletal pose. Step A is then repeated with the newly estimated lighting, which results in a better pose estimate. Steps A and B constitute the main part of our method and are described in Sec. 6.4. In Step C, we re-estimate incident lighting and surface albedo, and then refine the surface geometry. The refined surface now captures folds and bulges on the surface which cannot be modeled by articulated skeletal motion alone. For the initialization of the very first frame, we employ the same strategy as Gall *et al.* (2009) for pose estimation based on the manually segmented silhouettes. We also use a similar method as Chapter 5 to calculate the albedo value for each albedo segment, while the albedo segmentation could be provided by the user or any albedo segmentation method.

6.3 Image Formation Model

Similar to the previous chapters, we build upon the spherical harmonics (SH) parametrized reflectance equation, i.e. Eq. (2.5) and Eq. (2.7). We restate these two equations here for completeness:

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \sum_{l=0}^{N_D} \sum_{m=-l}^{l} L_{lm}\, T_{lm}(q) = k_d \boldsymbol{q} \sum_{k=1}^{n^2} l_k\, t_k(\boldsymbol{q}), \tag{6.1}$$

$$B(\boldsymbol{q}) = k_d(\boldsymbol{q}) \sum_{l=0}^{N_D} \sum_{m=-l}^{l} \Lambda_l\, g_{lm}\, \hat{\rho}_{dl}\, Y_{lm}(\boldsymbol{n}(\boldsymbol{q})), \tag{6.2}$$

where $B(\boldsymbol{q})$ is the reflected radiance on point $\boldsymbol{q}$, L_{lm} and l_k are the SH coefficients of the incident illumination, T_{lm} and t_k are SH coefficients for the combined visibility and clamped cosine function, k_d is the albedo value, N_D is the SH order employed and is equal to $n-1$, Λ_l is a constant scaling factor, $\hat{\rho}_d$ are SH coefficients for clamped cosine function, which are pre-given and constant, g_{lm} are the SH coefficients for the visible lighting term, and Y_{lm} is the SH basis function

which depends on the surface normal $\boldsymbol{n}(\boldsymbol{q})$. Similar to the previous chapters, our lighting estimation (Sec. 6.4.3) employs Eq. (6.1), and our shape refinement (Sec. 6.5) utilizes Eq. (6.2). Our major contribution is how we use Eq. (6.2) for skeletal pose estimation (Sec. 6.4), which we will elaborate on in the next section.

6.4 Pose Estimation Under Time-varying and Uncontrolled Illumination

At each time-step $t+1$, we perform a simultaneous estimation of body pose and incident lighting, both of which may have changed from time t. In order to keep the optimization tractable, we assume that changes in body pose are independent from changes in lighting, and alternate between the optimization of these variables.

We take as initialization the refined mesh and the embedded skeleton of time t, as well as the estimated incident lighting and surface albedo. In Sec. 6.4.1, we introduce how the mesh geometry changes according to the pose change. In Sec. 6.4.2, we define the shading constraint to estimate the pose parameters, given the incident lighting. The optimization to minimize the shading error is described afterwards. The method to estimate incident lighting is described in Sec. 6.4.3.

6.4.1 Surface Parameterization w.r.t. Pose

Image-based pose estimation methods usually need to model how the surface deforms with a change in pose. While some methods use a set of cylinders to approximate the human body, mesh skinning of a closed surface mesh usually gives a better approximation. Here we utilize the linear mesh skinning method Lewis *et al.* (2000) to deform the mesh to a skeletal pose (see Sec. 2.4.1). Supposing the position of vertex i to be $\boldsymbol{q}_i^t$ at time t, the new vertex position $\boldsymbol{q}_i^{t+1}$ at time $t+1$ is described by the following equation:

$$\begin{pmatrix} \boldsymbol{q}_i^{t+1} \\ 1 \end{pmatrix} = \sum_{i=1}^{m} w_i C_{J_i} \begin{pmatrix} \boldsymbol{q}_i^{t} \\ 1 \end{pmatrix}, \tag{6.3}$$

where C_{j_i} represents the rigid motion, including rotation and translation, of joint J_i, and w_i is the skinning weight, which connects the mesh surface to the embedded skeleton and can be computed from a standard toolbox Baran & Popović (2007). Similar to Bregler *et al.* (2004), we represent the articulated pose by a

set of twists $\theta_k\hat{\xi}_k$ (see Chapter 2). The state of a kinematic chain is determined by a global twist $\hat{\xi}$ and the joint angles $\boldsymbol{\Theta} = (\theta_1, \cdots, \theta_m)$. Assuming the state of the kinematic skeleton of the previous time-step to be known, the unknowns for pose estimation are the rigid motion of the root node and changes in joint angles which we denote as $\boldsymbol{\phi} = (\Delta\hat{\xi}, \Delta\theta_1, \cdots, \Delta\theta_m)$. Let $\boldsymbol{q}_i^t$ be the 3D position of vertex i at t. By using exponential maps to represent each joint's rigid motion and by linearizing the rigid body transforms, the position of the vertex i at $t+1$ can be expressed by the skinning equation as

$$\begin{aligned}\begin{pmatrix}\boldsymbol{q}_i^{t+1}\\1\end{pmatrix} &= \sum_{j=1}^{m} w_j e^{\Delta\hat{\xi}} \prod_{k\in R(j)} e^{\hat{\xi}_k\cdot\Delta\theta_k}\begin{pmatrix}\boldsymbol{q}_i^{t}\\1\end{pmatrix}\\ &\approx \begin{pmatrix}\boldsymbol{q}_i^{t}\\1\end{pmatrix} + \left(\Delta\hat{\xi} + \sum_{j=1}^{m} w_j \sum_{k\in R(j)} \hat{\xi}_k\cdot\Delta\theta_k\right)\begin{pmatrix}\boldsymbol{q}_i^{t}\\1\end{pmatrix} = \begin{pmatrix}\boldsymbol{q}_i^{t}\\1\end{pmatrix} + M_q(i)\cdot\boldsymbol{\phi},\end{aligned} \tag{6.4}$$

where $R(j)$ determines the indices of joints preceding the joint k in the kinematic chain. Each vertex i is assigned a set of skinning weights w_j that determine how much influence joint j has on the deformation of vertex i. Skinning weights are defined once during template building using standard techniques Baran & Popović (2007). $M_q(i)$ is the matrix determining how the pose change influences the change of vertex position, and has the form of

$$M_q(i) = \left[\boldsymbol{I}_{4\times 3},\quad -\hat{\boldsymbol{q}}_i^t,\quad W_{R(1)}\hat{\xi}_1\begin{pmatrix}\boldsymbol{q}_i^t\\1\end{pmatrix},\quad W_{R(2)}\hat{\xi}_2\begin{pmatrix}\boldsymbol{q}_i^t\\1\end{pmatrix},\quad \ldots,\quad W_{R(m)}\hat{\xi_m}\begin{pmatrix}\boldsymbol{q}_i^t\\1\end{pmatrix}\right], \tag{6.5}$$

where $W_{R(j)} = \sum_{k\in R(j)} w_k$ and is the sum of the skinning weights of vertices which are influenced by joint j, and $\hat{\boldsymbol{q}}_i^t$ is a skew-symmetric matrix and has the following form:

$$\hat{\boldsymbol{q}}_i^t = \begin{pmatrix}0 & -q_z & q_y\\ q_z & 0 & -q_x\\ -q_y & q_x & 0\\ 0 & 0 & 0\end{pmatrix}. \tag{6.6}$$

Eq. (6.4) can be rewritten as:

$$\begin{pmatrix}\Delta\boldsymbol{q}_t\\0\end{pmatrix} = \begin{pmatrix}\boldsymbol{q}_{t+1}\\1\end{pmatrix} - \begin{pmatrix}\boldsymbol{q}_t\\1\end{pmatrix} \approx \left(\Delta\hat{\xi} + \sum_{j=1}^{m} w_j \sum_{k\in R(j)} \hat{\xi}_k\cdot\theta_k\right)\begin{pmatrix}\boldsymbol{q}_t\\1\end{pmatrix} = M_q(i)\cdot\boldsymbol{\phi}. \tag{6.7}$$

A similar equation can be derived for the vertex normal $\boldsymbol{n}_i^{t+1}$ at time $t+1$:

$$\begin{pmatrix} \boldsymbol{n}_i^{t+1} \\ 0 \end{pmatrix} \approx \begin{pmatrix} \boldsymbol{n}_i^{t} \\ 0 \end{pmatrix} + M_n(i) \cdot \boldsymbol{\phi}, \tag{6.8}$$

where $M_n(i)$ is a matrix that determines how a change in pose is related to a change in normal orientation.

6.4.2 Shading Constraint for Pose Estimation

Our shading constraint requires the rendered images of the optimal pose according to our image formation model to be as close as possible to the image data captured. Following Eq. (6.2), the shading constraint for a single camera c can be defined as

$$E_c^s = \sum_i (k_d(i) \sum_{l=0}^{N_D} \sum_{m=-l}^{l} \Lambda_l \, \hat{\rho}_{dl} \, g_{lm} \, Y_{lm}(\boldsymbol{n}_i) - I_c^{t+1}(x_i^{t+1}, y_i^{t+1}))^2, \tag{6.9}$$

where (x_i^{t+1}, y_i^{t+1}) is the projection of the surface vertex $\boldsymbol{q}_i^{t+1}$, and other symbols are defined similarly to Eq. (6.2). We assume the albedo $k_d(i)$ at time $t+1$ is the same as that at time t, thereby exploiting temporal coherence in scene motion. However, both the lighting and geometry at time $t+1$ are unknown, in other words g_{lm} and $\boldsymbol{n}$ are unknown. We attempt to estimate both of them in a unified framework in order to properly account for shading changes due to changes in either lighting or pose. Since simultaneous estimation of both of them is computationally challenging and less stable, we alternate between error minimization with respect to each of these two variables. Firstly, we minimize the shading error to estimate the pose, by assuming the lighting to be the same as in the previous time-step, and thereafter we solve for the new lighting. To do this, we linearize the SH term $Y_{lm}(\boldsymbol{n}_i^{t+1})$ and the image intensity term I_c^{t+1} from Eq. (6.9). The SH term is expressed in a first-order Taylor-series expansion, using Eq. (6.8), as follows:

$$Y_{lm}(\boldsymbol{n}_i^{t+1}) \approx Y_{lm}(\boldsymbol{n}_i^{t}) + \frac{\partial Y_{lm}(\boldsymbol{n}_i^{t})}{\partial \boldsymbol{n}_i^{t}} \Delta \boldsymbol{n}_i^{t} = Y_{lm}(\boldsymbol{n}_i^{t}) + \frac{\partial Y_{lm}(\boldsymbol{n}_i^{t})}{\partial \boldsymbol{n}_i^{t}} M_n(i) \cdot \boldsymbol{\phi}, \tag{6.10}$$

where $\frac{\partial Y_{lm}(\boldsymbol{n}_i^{t})}{\partial \boldsymbol{n}_i^{t}}$ is the derivative of the SH basis function with respect to normal changes $\Delta \boldsymbol{n}_i^{t}$, which can be expressed in terms of pose changes $\boldsymbol{\phi}$.

Similar to the computation of optical flow Brox *et al.* (2004), we linearize $I^{t+1}(x_i^{t+1}, y_i^{t+1})$ as:

$$I^{t+1}(x_i^{t+1}, y_i^{t+1}) = I^{t+1}(x_i^t + u_i, y_i^t + v_i) \approx I^{t+1}(x_i^t, y_i^t) + I_x^{t+1} u_i + I_y^{t+1} v_i. \quad (6.11)$$

Here, (u_i, v_i) is the 2D flow, i.e. the displacement in the image due to the motion of vertex $\boldsymbol{q}_i$. Next, we derive the linear approximation for the flow (u_i, v_i) in an image from the motion parameters $\boldsymbol{\phi}$. This is similar to the derivation in Bregler *et al.* (2004), but we use the full perspective camera model instead of scaled orthographic projection Bregler *et al.* (2004), as camera calibration is available in our system. In detail, the following equation describes the projected location (x_i^{t+1}, y_i^{t+1}) of a 3D point $\boldsymbol{q}_i^{t+1}$, whose position is determined by skeletal position $\boldsymbol{\phi}$ by means of a perspective camera model:

$$\begin{pmatrix} x_i^{t+1} \\ y_i^{t+1} \end{pmatrix} = \begin{pmatrix} \frac{s_1}{Z_i^{t+1}} & 0 & 0 & u_0 \\ 0 & \frac{s_2}{Z_i^{t+1}} & 0 & v_0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^{t+1} \\ 1 \end{pmatrix}, \quad (6.12)$$

where s_1, s_2 are the focal length in x axis and y axis respectively, (u_0, v_0) is the principle point, Z_i^{t+1} is the depth of $\boldsymbol{q}_i^{t+1}$ for the current camera, and $\boldsymbol{e}^{\hat{\xi}_c}$ is the extrinsic matrix, i.e. the camera's pose. Then, the image motion from time t to time $t+1$ can be expressed as follows:

$$\begin{aligned}
\begin{pmatrix} u_i \\ v_i \end{pmatrix} &= \begin{pmatrix} \frac{s_1}{Z_i^{t+1}} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^{t+1}} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^t \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{s_1}{Z_i^t - \Delta Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t - \Delta Z_i^t} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^t \\ 1 \end{pmatrix} \\
&\approx \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \left(\begin{pmatrix} \boldsymbol{q}_i^{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} \boldsymbol{q}_i^t \\ 1 \end{pmatrix} \right) + \begin{pmatrix} \frac{s_1 \Delta Z_i^t}{Z_i^{t\,2}} & 0 & 0 & 0 \\ 0 & \frac{s_2 \Delta Z_i^t}{Z_i^{t\,2}} & 0 & 0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^{t+1} \\ 1 \end{pmatrix} \\
&\approx \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z^t} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \Delta \boldsymbol{q}_i^t \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{s_1 \Delta Z_i^t}{Z_i^{t\,2}} & 0 & 0 & 0 \\ 0 & \frac{s_2 \Delta Z_i^t}{Z_i^{t\,2}} & 0 & 0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \left(\begin{pmatrix} \boldsymbol{q}_i^t \\ 1 \end{pmatrix} + \begin{pmatrix} \Delta \boldsymbol{q}_i^t \\ 0 \end{pmatrix} \right) \\
&\approx \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \Delta \boldsymbol{q}_i^t \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{s_1}{Z_i^{t\,2}} & 0 & 0 & 0 \\ 0 & \frac{s_2}{Z_i^{t\,2}} & 0 & 0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi}_c} \cdot \begin{pmatrix} \boldsymbol{q}_i^t \\ 1 \end{pmatrix} \cdot \Delta Z_i^t.
\end{aligned} \quad (6.13)$$

The linearization is based on the assumption that the rigid motion $\Delta \boldsymbol{q}_i^t$ as well as the relative depth change $\Delta Z_i / Z_i^t$ are small. The depth change ΔZ_i^t can be

further expressed through the motion parameters:

$$\begin{aligned}\Delta Z_i^t &= -\left[\boldsymbol{e}^{\hat{\xi}_c}\begin{pmatrix}\Delta \boldsymbol{q}_i^t \\ 0\end{pmatrix}\right]_z = -\left[\boldsymbol{e}^{\hat{\xi}_c}\cdot M_q(i)\cdot \boldsymbol{\phi}\right]_z \\ &= -\left[\begin{pmatrix}\boldsymbol{r}_1^T & t_1 \\ \boldsymbol{r}_2^T & t_2 \\ \boldsymbol{r}_3^T & t_3\end{pmatrix}\cdot M_q(i)\cdot \boldsymbol{\phi}\right]_z = -\begin{bmatrix}\boldsymbol{r}_3^T & t_3\end{bmatrix}\cdot M_q(i)\cdot \boldsymbol{\phi},\end{aligned} \tag{6.14}$$

where $\boldsymbol{r}_3^T$ is the $3rd$ row of the rotation matrix of the camera pose. As the $4th$ row of $M_q(i)$ contains only zeros, t_3 can be omitted in the above equation. Considering this, the flow (u_i, v_i) can be expressed as a linear function of the pose change $\boldsymbol{\phi}$ as follows:

$$\begin{aligned}\begin{pmatrix}u_i \\ v_i\end{pmatrix} &\approx \begin{pmatrix}\frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4\end{pmatrix}\cdot \boldsymbol{e}^{\hat{\xi}_c}\cdot\begin{pmatrix}\Delta \boldsymbol{q}_i^t \\ 0\end{pmatrix} + \begin{pmatrix}\frac{s_1}{Z_i^{t2}} & 0 & 0 & 0 \\ 0 & \frac{s_2}{Z_i^{t2}} & 0 & 0\end{pmatrix}\cdot \boldsymbol{e}^{\hat{\xi}_c}\cdot\begin{pmatrix}\boldsymbol{q}_i^t \\ 1\end{pmatrix}\cdot \Delta Z_i^t \\ &\approx \left\{\begin{pmatrix}\frac{s_1}{Z_i^t} & 0 & 0 & s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4\end{pmatrix}\boldsymbol{e}^{\hat{\xi}_c} - \begin{pmatrix}\frac{s_1}{Z_i^{t2}} & 0 & 0 & 0 \\ 0 & \frac{s_2}{Z_i^{t2}} & 0 & 0\end{pmatrix}\boldsymbol{e}^{\hat{\xi}_c}\begin{bmatrix}\boldsymbol{q}_i^t \\ 1\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{r}_3^T & 0\end{bmatrix}\right\}\cdot M_q(i)\cdot\boldsymbol{\phi},\end{aligned} \tag{6.15}$$

The shading constraint in Eq. (6.9) can be further improved by considering the color similarity between the rendered color and the image color. This color similarity is computed as the Euclidean distance in HSV space and appears as a weighting factor μ_i in our shading constraint. This helps us avoid optimizing the model where the template material does not yet match the material visible in the projected region of an input image. Combining terms from multiple cameras, our non-linear multi-view shading energy function is then given as

$$E = \frac{1}{N_c}\sum_c\sum_i\{\mu_i^c\,(k_d(i)\sum_{l=0}^{N_D}\sum_{m=-l}^{l}\Lambda_l\,\hat{\rho}_{dl}\,g_{lm}\,Y_{lm}(\boldsymbol{n}_i) - I_c^{t+1}(x_i^{t+1}, y_i^{t+1}))\}^2, \tag{6.16}$$

where N_c is the total number of constraints for error normalization (*i.e,* the number of pixels in which a mesh surface is visible), and μ_i^c is the color similarity for pixel i in camera c. The non-linear energy defined in Eq. (6.16) will be solved by an iterative solver that minimizes the energy in a sequence of linearized versions of the original nonlinear problem. In order to do this, we use the previously explained recipe for linearization, and can express Eq. (6.16) as a linear system depending on a small change in pose parameters $\boldsymbol{\phi}$ as follows:

$$H\cdot\boldsymbol{\phi} = \boldsymbol{b}. \tag{6.17}$$

Here, the k^{th} rows of matrix H and vector $\boldsymbol{b}$ have the following form:

$$\begin{aligned} H_k = \mu_i^c k_d(i)(\sum_{l=0}^{N_D}\sum_{m=-l}^{l}\Lambda_l\hat{\rho}_{dl}g_{lm}\cdot\frac{\partial Y_{lm}(\boldsymbol{n}_i^t)}{\partial \boldsymbol{n}_i^t})M_n(i) \\ -\mu_i^c\left[\tfrac{s_1}{Z_i^t}I_x^{t+1},\quad \tfrac{s_2}{Z_i^t}I_y^{t+1},\quad 0,\quad s_3I_x^{t+1}+s_4I_y^{t+1}\right]\boldsymbol{e}^{\hat{\xi_c}}M_q(i) \\ +\mu_i^c\left[\tfrac{s_1}{Z_i^{t2}}I_x^{t+1},\quad \tfrac{s_1}{Z_i^{t2}}I_y^{t+1},\quad 0,\quad 0\right]\boldsymbol{e}^{\hat{\xi_c}}\begin{bmatrix}\boldsymbol{q}_i^t\\1\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{r}_3^T & 0\end{bmatrix}\cdot M_q(i), \\ \boldsymbol{b}_k = \mu_i^c I^{t+1}(x_i^t,y_i^t) - \mu_i^c k_d(i)\sum_{l=0}^{N_D}\sum_{m=-l}^{l}\Lambda_l\hat{\rho}_{dl}g_{lm}Y_{lm}(\boldsymbol{n}_i^t). \end{aligned} \tag{6.18}$$

Coarse-to-fine Optimization To minimize the non-linear error function of Eq. (6.16), we iteratively solve Eq. (6.17) and linearize around the new solution. Note that here after solving Eq. (6.17), we check if the original energy in Eq. (6.16) decreases to decide the appropriate step size for updating the solution, in a fashion similar to Newton-Raphson style minimization with adaptive step size. The linearization as given in Eq. (6.11) assumes that the local image intensity variations can be approximated by a first-order Taylor expansion. So we adopt a coarse-to-fine strategy for pose estimation — by building an image pyramid through successively downsampling each captured image, and running the pose estimation from the coarsest images to the finest images. This helps us to properly track faster motions with bigger image displacements, and reduces the chance of getting stuck in local minima.

6.4.3 Lighting Optimization

In general, lighting changes can be abrupt, and severe lighting changes, that occur within two consecutive frames of video, are difficult to model. However, for most cases, it can be assumed that the lighting at $t+1$ changes gradually from lighting at t. In our method, we optimize for pose and lighting in a two pass strategy. For the first pass, we use the lighting at t to optimize for pose at $t+1$, as described in the previous section. For the second pass, we estimate the lighting at $t+1$ based on the new pose, and then use it to refine the pose estimates. We have empirically observed that one additional iteration of alternating optimization is sufficient for getting good estimates.

We derive the constraint for lighting optimization from the image formation model defined in Eq. (6.1). Similar to Chapters 4 and 5, we compare the rendered intensity values with the captured image I_c and estimate the lighting coefficients l_k by solving an ℓ_1 norm minimization problem defined as:

$$\hat{l} = \underset{l}{\operatorname{argmin}} \sum_{i} \sum_{c \in Q(i)} \left| k_d(q_i) \sum_{k=1}^{n^2} l_k \, t_k - I_c(P_c(\boldsymbol{q}_i)) \right|. \tag{6.19}$$

Here, i is the vertex index, c is the camera index, $Q(i)$ is the set of cameras that can see the i-th vertex $\boldsymbol{q}_i$, P_c is the projection matrix for camera c, k_d is the albedo value, t_k are the SH coefficients for the combined visibility and clamped cosine function, and $n-1$ is equal to the SH order employed.

6.5 Dynamic Surface Refinement

After the pose and lighting estimation step, we have a coarse template model that strikes the correct pose, as parameterized by the respective skeleton pose parameters. We here use quaternion blend skinning Kavan *et al.* (2007) to obtain the final shape of the surface mesh in the current pose, as it leads to higher quality surface deformation (see Sec. 2.4.2). Before proceeding with the shading-based shape refinement, we need to first acquire the reflectance for each vertex, i.e. the albedo value. Similar to Chapter 5, we represent the surface reflectance on the mesh by a set of albedo segments, while the vertices in each segment share the same albedo value. As our reconstructed mesh for each frame is temporally coherent, and also the albedo of each vertex is known for the previous frame, we can readily obtain the albedo value for the mesh of the current frame. However, due to potential shifting of the garment, the albedo segmentation of the previous frame may not be correct for the current frame, especially for the vertices located on the boundary of the segments. Thus, in this chapter we update the albedo segmentation in a temporally coherent way. By formulating the segmentation as a Markov Random Field (MRF) problem, our method is able to generate a segmentation which uses a consistent set of materials for each time-step, to preserve boundaries between materials on the surface, and to represent the potential shifting of material over the surface, which can be caused for example by sliding apparel. In detail, we assume the surface albedos belong to a set of K distinct materials $\{d_1^t, \ldots, d_K^t\}$ and want to determine the albedo labels $\boldsymbol{a^t} \in \{1, \ldots, K\}$ of the vertices. Assuming the material segmentation $\boldsymbol{a^t}$ at frame t is given, we

segment the mesh at frame $t+1$ by finding the lowest energy configuration of the MRF defined as:

$$\psi(\boldsymbol{a^{t+1}}) = \sum_{i=1}^{N_q}(\phi(O|a_i^{t+1}) + \sum_{j\in N_r(i)} \phi(a_i^{t+1}, a_j^{t+1})), \tag{6.20}$$

where N_q is the number of vertices on the mesh, $N_r(i)$ is the neighboring vertex set of vertex i, $\phi(a_i^{t+1}, a_j^{t+1})$ is a smoothness term that takes the form of a generalized Potts model Szeliski *et al.* (2008), and $\phi(O|a_i^{t+1})$ is a likelihood data term which imposes individual penalties for assigning an albedo label to vertex i according to the observation O. The data term combines two terms. The first is the color prior term for each segment:

$$\phi_c\left(O|a_i\right) = \left(o_i^{t+1} - d^t_{a_i^{t+1}}\right)^2, \tag{6.21}$$

where o_i^{t+1} is the initial albedo estimate of vertex i at frame $t+1$ based on captured image irradiance and the estimated lighting $\boldsymbol{l^{t+1}}$ from Sec. 6.4.3. The second term is the albedo label prior, which penalizes different labels in consecutive time-steps:

$$\phi_s\left(O|a_i^{t+1}\right) = \begin{cases} C_P, & if\ a_i^{t+1} \neq a_i^t \\ 0, & if\ a_i^{t+1} = a_i^t \end{cases} \tag{6.22}$$

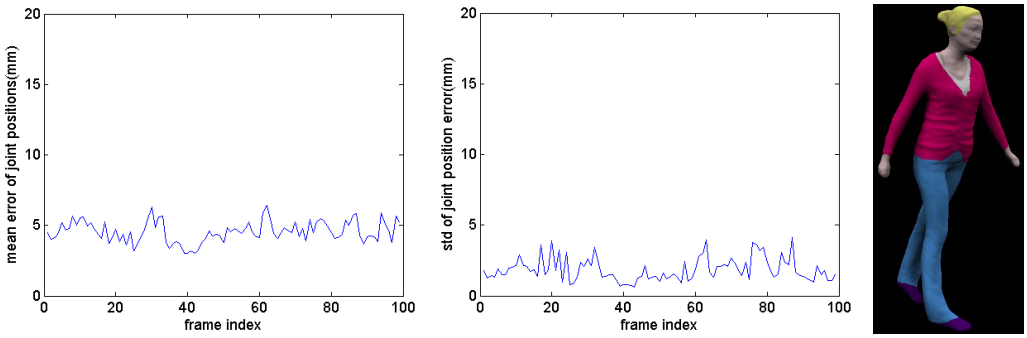
where C_P is a preset penalty constant. Finally, the MAP-MRF energy function defined in Eq. 6.20 is minimized via graph cuts Boykov & Funka-Lea (2006); Szeliski *et al.* (2008).

When the albedo segmentation is updated, namely the vertex albedo is obtained for time $t+1$, we refine the vertex position $\boldsymbol{q}_i^{t+1}$ on the coarse mesh from shading cues using the method described in Chapter 5. With this step, we are able to capture the non-rigid surface deformation, for instance the folds and wrinkles, which are not being captured using the skeletal motion tracking.

6.6 Results

6.6.1 Quantitative Evaluation

In order to quantitatively evaluate our method, we generated a synthetic sequence of 100 frames with 10 camera views. The ground-truth skeleton and mesh geometry are taken from the captured results of the human walking sequence that we reconstruct in Chapter 5. The ground-truth surface albedo map and dynamically



(a) (b) (c)

Figure 6.3: Quantitative evaluation: (a) the mean error of joint positions, (b) the standard deviation of joint position errors, (c) a generated synthetic image.

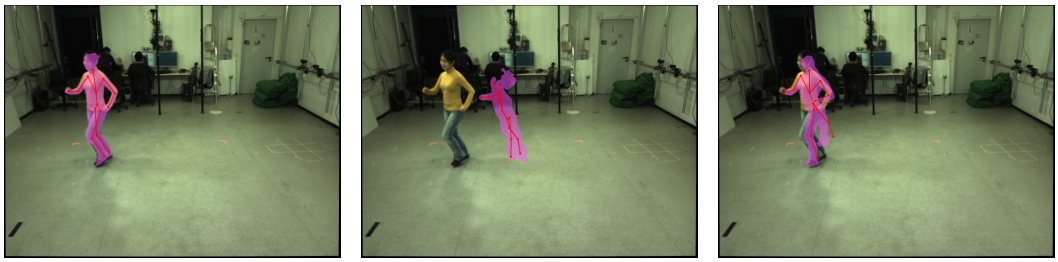
changing illumination are manually assigned. With these generated synthetic images as input, and given the mesh, skeleton, and albedo segmentations for the first frame, we run our algorithm on the remaining 99 frames. In Fig. 6.3, we report the accuracy of our approach. The error metric used is the difference in 3D positions between the ground truth and the reconstructed joint positions. The mean joint position error is about 6 mm, which shows the high accuracy of our method.

6.6.2 Real-world Sequences

We use three real captured sequences for qualitatively evaluating our method. The sequences were captured with 11 cameras in a studio. The subject can wear sparsely textured apparel, such as a t-shirt or a sweater with a simple color. But unlike in the input data of previous performance capture methods, there is no need for a green-screen background, and there may be potentially occluding objects in the scene and dynamic background (Fig. 6.1). Cameras recorded at a resolution of 1296×972 pixels, and at a frame rate of $40 fps$. Each sequence shows major illumination changes; they are induced by an operator randomly setting control knobs for various lights in the studio — these readings are not taken nor provided in any way to our method. Please also note that some of the captured images are saturated, which our method handles robustly. As can be seen in the overlaid images of our estimated skeleton and 3D shape in Fig. 6.4, good pose estimates are obtained despite the challenging scene conditions. Even when a few cameras are partially occluded, our method still works quite well owing to the

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

Figure 6.4: Illumination changes in a real captured sequence: (a,c) frames showing widely differing incident illumination; (b,d) the output skeletal pose and mesh overlaid onto the images. The insets show estimated illumination at each frame.



(a) (b) (c)

Figure 6.5: Comparison with alternative tracking methods: (a) our method, (b) texture-based tracking, (c) silhouette-based tracking Gall *et al.* (2009)

use of shading cues and the multiple camera setup. High quality surface details, such as deforming cloth folds, are also captured (Fig. 6.6).

We compare the results of our method with a texture-based tracker that does not estimate lighting explicitly at each frame. Instead, it assumes texture from the first frame and uses optical flow for tracking; it loses tracking after a few frames as the lighting changes significantly (see Fig. 6.5-b). We also tested against a silhouette-based tracker Gall *et al.* (2009) that explicitly performs background segmentation using chroma-keying on the captured images. Due to changing lighting, the extracted silhouettes are sometimes misleading and result in inaccurate pose estimates (see Fig. 6.5-c).

6.6.3 Computation Time

The computation time of our method depends on image resolution, mesh resolution, and the level of detail at which the lighting components are modeled, in particular the order of spherical harmonics that is used. In our experiments, we

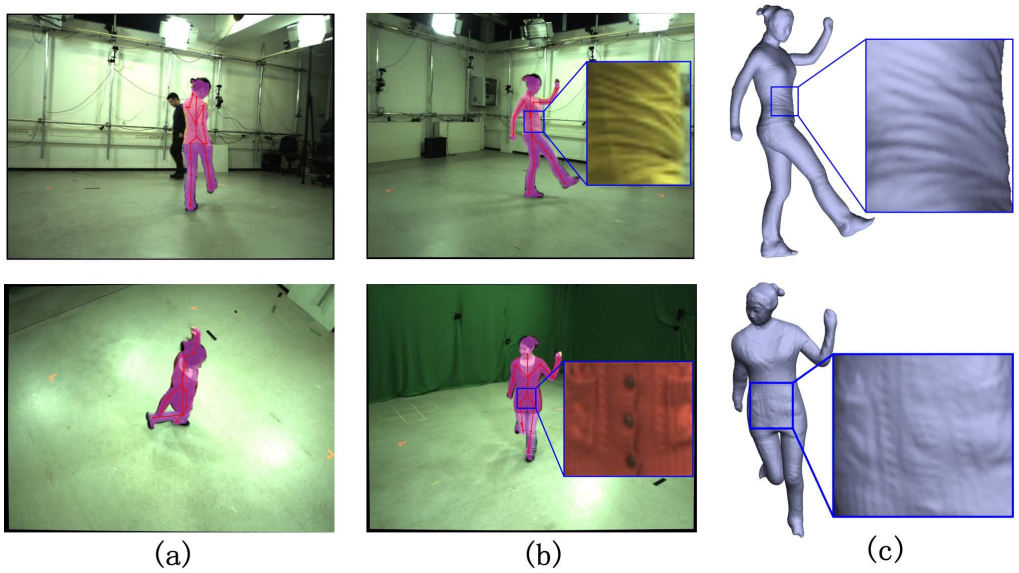


Figure 6.6: Results of pose and 3D shape estimation: (a,b) overlaid skeletal pose at different frames and camera views, (c) fine-scale 3D shape reconstruction. The inset shows dynamic cloth deformations captured from shading.

represented 3D shape using meshes of 80000 vertices, and used a 4th order SH for representing lighting. With these values, our method takes about 10 minutes per frame on a standard CPU with a 2.6 GHz processor and 8 GB RAM. Specifically, the computation times are 3 minutes for one pass of pose estimation, which we do twice for each frame. The lighting estimation step is quite fast, taking only 10 seconds. The other time-consuming part is the dynamic shape refinement, which takes 4 minutes, of which 1 minute is spent on visibility calculation. Striking a trade-off between representation accuracy and computation time, we utilized a low-resolution mesh (around 5000 vertices) to render the visibility map for each vertex on the high-resolution mesh. As our code is unoptimized, we believe the computational time can be further reduced by parallelizing the algorithm.

6.6.4 Discussion

Our assumptions of Lambertian reflectance and a local shading model may not be justified in some cases, for instance scenes with strong specularity, etc. Abrupt lighting changes, e.g., in a disco, or the illumination generated by a controlled light stage, are also hard to model. However, in cases where the lighting pattern is known, it can be directly provided as input to our method. Since we estimate

lighting and pose sequentially at each time-step, error accumulation may cause drift of the tracker. In future work, we would like to address this issue by stronger motion priors that can be learned from data. A final limitation is the computation time for running our method, which is too high for real-time deployment. We would like to address these and other limitations in future work.

6.7 Conclusion

In this chapter, we provide a novel shading based framework for human performance capture under uncontrolled and dynamic lighting. Starting from synchronized multi-view images, we estimate both the articulated human pose and fine-scale time-varying surface geometry. A key innovation is a novel iterative pose optimization framework developed based on the inverse rendering. Our approach does not expect carefully engineered backgrounds as it does not perform silhouette extraction or any other form of background segmentation. Thus, our approach is not limited to the specified lighting and background, and can be applied to more general scenarios, where the lighting is general, unknown and even time-varying. Still, our method requires many cameras for recording. In the next part of this thesis, we will describe algorithms that are able to work with a low number of recording cameras.

6.7 Conclusion

Part III

Binocular Performance Capture

With a controlled studio setup, marker-less performance capture are able to produce accurate motion and high-quality dynamic geometry. However, the required capture setup is still very constraining and prevents these methods from being applied to general scenarios, e.g. a real movie set. In the previous part, we showed how to relax these constraints by investigating inverse rendering, enabling marker-less performance capture under general lighting and with less-constrained backgrounds. However, it still requires many cameras for the capture, and it is not trivial to set up a multi-view system on a real movie set. Thus, in this part we focus on reducing the required number of cameras for performance capture to as few as two cameras, i.e., a stereo camera setup, by resorting to inverse rendering concepts. This setup mimics real movie production setups, where a primary stereo camera is often available. In Chapter 7, we develop a new method to capture facial performance under uncontrolled illumination by using a single pair of cameras. In Chapter 8, a new performance capture method is proposed to capture the full-body performance of multiple actors on set by employing a stereo camera.

Chapter 7

Binocular Facial Performance Capture under Uncontrolled Lighting

7.1 Introduction

A convincingly rendered face and a convincingly animated facial performance are the essential features of realistic virtual actors. To achieve this, high quality dynamic facial geometry is required. To meet these high quality demands, the research community has developed a variety of facial performance capture techniques which aim to reconstruct very detailed dynamic facial geometry, motion, and possibly appearance from sensor measurements of real subjects. On the one hand, there are active optical systems that use markers, active illumination, or invisible paint to capture facial performance Bickel *et al.* (2007); Furukawa & Ponce (2009); Zhang *et al.* (2004). However, such reconstructions often lack detail and appearance capture is difficult or impossible. On the other hand, passive approaches use multiple cameras and vision-based reconstruction techniques to capture facial performance, e.g., Bradley *et al.* (2010). Reconstructions are of high quality, but pore-level detail is often missing. Moreover, accumulating drift makes it hard to capture very expressive motion. Active lighting methods can bring out pore-level shape detail, but the price to be paid is a complex controlled light and camera setup Vogiatzis & Hernández (2011); Wilson *et al.* (2010). In other words, to capture facial performance with high-quality spatial and temporal detail, current state-of-the-art techniques require a large number of cameras in a controlled indoor environment, possibly actively controlled illumination, and in

many cases some form of active interference with the scene.

These strong requirements are the reason why high-quality facial performance capture has so far mostly been a privilege of high-budget animation productions. In addition, facial performance capture has scarcely been used where it would actually be most effective: in arbitrary uncontrolled settings, such as indoor and outdoor movie sets where the actors perform in their natural environment. Also, in light of an ever-growing amount of existing stereo movie footage, a method that just makes use of the principal stereo camera could serve as a cornerstone for new movie-production applications, such as facial performance capture from the principal stereo camera feed.

In this chapter, we take a step towards the goal of lightweight facial performance capture in a general environment by proposing a new image-based facial performance capture approach that uses a single stereo pair of video cameras. It succeeds under uncontrolled and time-varying illumination, either indoors or outdoors. It allows both the performer and the stereo pair of cameras to move independently and yields detailed 3D facial performance geometry that is fully spatio-temporally coherent, even when performers have very expressive faces. We show highly detailed 3D facial performance results, optionally with texture, that were reconstructed under uncontrolled and time-varying lighting with two different camera systems: results from a pair of SLR cameras capturing indoors, and results of previously unseen detail captured outdoors with a low quality consumer stereo camera (see Sec. 7.6). The work presented here was published in Valgaerts *et al.* (2012b).

7.2 Method Overview

As input, our approach expects a stereo video sequence of a face captured in a general environment. Our method is composed of two main computational pipelines (Fig. 7.1):

I In a first pass, we track a coarse detail face template throughout a binocular stereo sequence. This *template tracking step* (Sec. 7.4) produces a sequence of coarse face meshes that are in full correspondence and show minimal drift. To enable this, our approach makes use of a highly accurate image-based scene flow method and relies on a Laplacian deformation model to regularize the moving geometry.

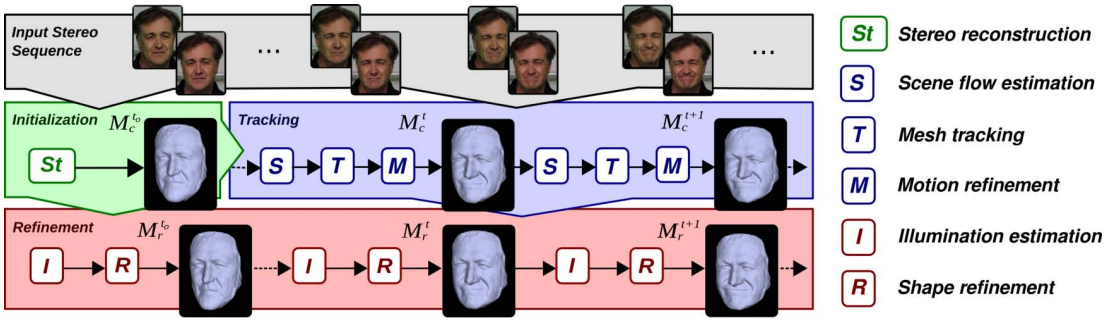


Figure 7.1: Overview of our facial performance capture method with two processing pipelines. Starting from an initial stereo reconstruction, a coarse template mesh is tracked in a first pass. In a second pass, the tracked coarse geometry is refined using shading information.

II In a second pass, we add fine time-varying detail, e.g., wrinkles and folds, to the tracked meshes. This *shape refinement step* (Sec. 7.5) exploits shading information to produce accurate surface detail under general and changing lighting. We build upon a framework for incident lighting and albedo estimation, and contribute with both a new albedo clustering approach, and an improved shape refinement optimization that produces better results with an order of magnitude lower computation time than the method proposed in Chapter 5.

Thus, we capture facial performance in a *coarse-to-fine manner*: While the first pass is responsible for the recovery of coarse-scale head motion and facial deformation, the second pass refines the results to include fine-scale detail at skin level.

In the next sections we will discuss both pipelines in detail. Henceforth, we will indicate by I_0^t the left frame of a binocular stereo sequence at time t, and by I_1^t the corresponding right frame. Without loss of generality, we can assume that I_0^t and I_0^{t+1} (I_1^t and I_1^{t+1}) are two consecutive frames in the left (right) image sequence for any time t. We further denote by t_0 the time at which we start capturing, i.e., $(I_0^{t_0}, I_1^{t_0})$ is the first stereo pair in our tracking and refinement algorithm. A reconstructed triangular mesh at time t will be denoted by M^t. The Euclidean coordinates of a vertex at time t will be denoted by $\boldsymbol{q}^t$. Please note that our two pipelines reconstruct a coarse mesh M_c^t and a refined mesh M_r^t at each time step, both of which are based on the same vertex set and connectivity.

7.3 Initialization

We assume that the stereo camera pair is calibrated off-line (MATLAB toolbox[1]). Our method starts off from a smooth 3D reconstruction of the face that will serve as a *template mesh* for the tracking step. During mesh tracking, this template will be moved and deformed according to the detected motion in the stereo sequence.

Template Reconstruction It is assumed that the face at time t_0 is at rest. To obtain an initial 3D reconstruction from the first stereo pair $(I_0^{t_0}, I_1^{t_0})$, we apply a variant of the variational stereo method of Valgaerts *et al.* (2012a) for calibrated images. This method recovers the dense 2D displacement field between $I_0^{t_0}$ and $I_1^{t_0}$ by minimizing an energy function of the form:

$$E = \int_\Omega \left(E_\mathrm{D} + \alpha\, E_\mathrm{G} + \beta\, E_\mathrm{S}\right) \mathrm{d}\boldsymbol{x} \ , \tag{7.1}$$

where E_D imposes constancy assumptions on certain image features, E_G includes knowledge about the known stereo geometry, and E_S assumes the displacement field to be piecewise smooth. The mathematical form of these terms can be found in the equations (2.15), (2.17) and (2.19) respectively. We also use a similar minimization method to that used in the scene flow estimation. Please refer to Sec. 2.2 for details.

Once the 2D displacement field has been recovered, the corresponding pixels can be triangulated to obtain a 3D point cloud Hartley & Zisserman (2000). In practice, we perform a 3D reconstruction for both pairs $(I_0^{t_0}, I_1^{t_0})$ and $(I_1^{t_0}, I_0^{t_0})$. This ensures a sufficient number of 3D points in regions that are poorly visible in just one image, such as the sides of the nose.

In a post-processing step, the background is removed manually and the point cloud is converted to a triangular mesh Kazhdan *et al.* (2006). We set the number of vertices roughly equal to that of the pixels in the face region such that each vertex corresponds to a pixel in the input views. Finally, the mesh is smoothed Sorkine (2005) and each vertex is assigned a fixed color using projective texturing and blending from both input views. If desired, holes can be cut in the the mesh for the mouth or the eyes.

The above steps are illustrated in Fig. 7.2, where we show the starting frames $I_0^{t_0}$ and $I_1^{t_0}$ of a stereo sequence, together with the obtained 3D reconstruction and the final template mesh $M_c^{t_0}$.

[1] www.vision.caltech.edu/bouguetj/calib_doc/

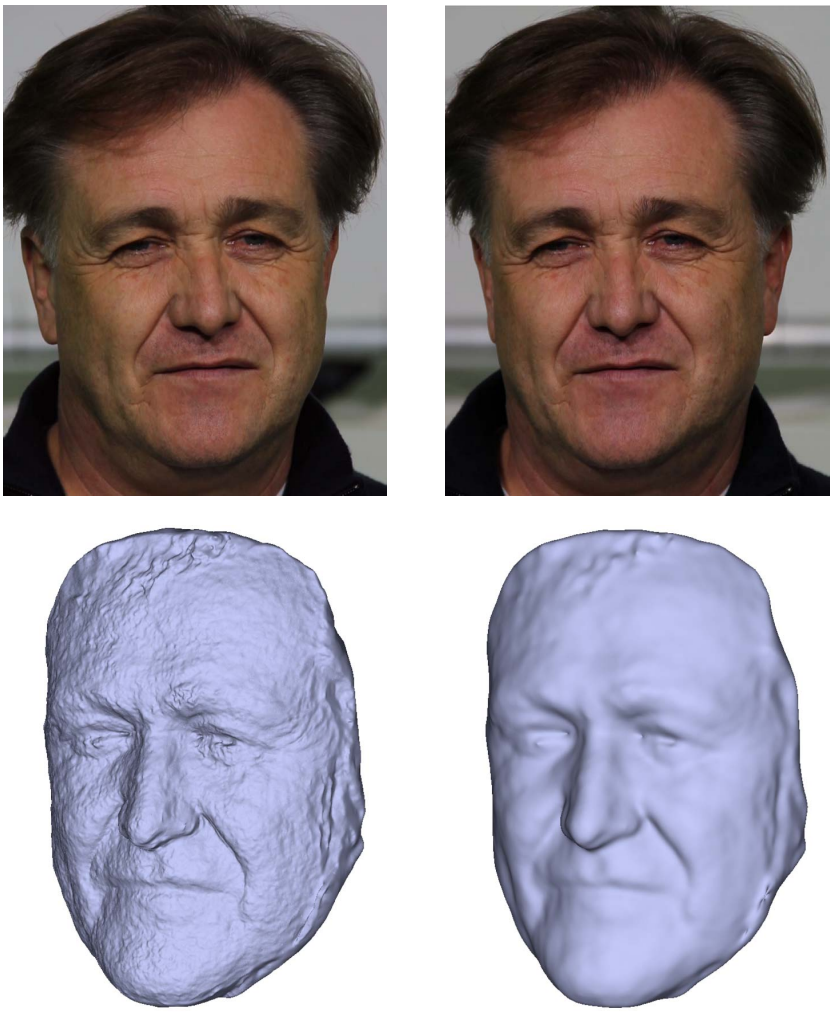


Figure 7.2: Initialization. Top row: starting frames $I_0^{t_0}$ and $I_1^{t_0}$. Bottom row: stereo reconstruction and template mesh $M_c^{t_0}$.

7.4 Template Tracking

The tracking step is responsible for propagating the template mesh throughout the stereo sequence. To accurately recover the motion of the face, we base our tracking on a state-of-the-art method for scene flow computation, which is introduced in Sec. 2.2. This method establishes a dense 3D displacement field, which is used to update the position of all the vertices in the tracked mesh from one time instance to the next. A smooth deformation of the face is obtained by regularizing the geometry of the surface via the Laplacian operator. Scene flow estimation is then used a second time to refine the motion and to minimize any reprojection error that might have been induced by the tracking.

All corresponding pixels can be triangulated to obtain a 3D reconstruction and a 3D displacement field. Note that we have only used 2D optical flow as an intermediate representation during the scene flow estimation. Our tracking algorithm effectively uses a 3D motion field, where each scene flow vector is characterized by a 3D starting position $\boldsymbol{s}^t$ and a 3D vector $\boldsymbol{d}^t$. Note that we are able to cope with large motion and even noticeable motion blur.

7.4.1 Mesh Tracking

Once the scene flow $\boldsymbol{d}^t$ has been estimated for time instance t, it can be used to propagate the vertices of the current coarse mesh M_c^t to their new positions at t+1. However, moving each vertex by its corresponding scene flow vector is likely to induce local drift, which would quickly destroy the integrity of the template mesh. The reason for this is that the computed scene structure and motion contain errors, e.g., due to noise, which cannot be completely removed by our scene flow regularization. In addition, our scene flow lacks temporal coherence because it is estimated independently on all time instances. To ensure that the tracked geometry remains smooth over time, we have to regularize the moving geometry.

Positional Constraints To preserve the smoothness of the tracked mesh, we only assign a scene flow vector to a subset C^t of vertices. These vertices will be denoted as *constrained vertices*, because their locations will form the positional constraints in the regularization of the mesh geometry. Here we select the constrained vertices uniformly over the mesh to ensure a sufficient distribution of positional constraints. Additionally, we ensure on each time instance t that

all vertices in C^t are visible in both the left and the right image. This avoids erroneous tracking in regions that become occluded by, e.g., head movement or expressive facial motion. For some outdoor sequences, we experienced strong interference of the estimated background motion at the side of the face. To avoid drift in more oblique regions, we restrict C^t for such cases to vertices for which the angle between the surface normal and the camera's principal axis lies below a certain threshold ($\leq 70°$).

Positional Update A straightforward way of updating the position of a constrained vertex $\boldsymbol{q}_i^t$, $i \in C^t$, would be to move it to the end point of the closest scene flow vector, thus calculating its new position as $\boldsymbol{s}_i^t + \boldsymbol{d}_i^t$. This strategy effectively fits the mesh M_c^t to the target point cloud computed by our scene flow algorithm, but would only make sense if both the 3D reconstruction and the 3D displacement field are estimated with equally high accuracy. In practice, however, the estimated 3D structure is noisier than the scene flow because the optical flow due to the change in view point is generally larger than the flow due to face motion. Also, possible differences between the left and right camera can play a role. Especially for outdoor capture from low quality cameras, this strategy led to undesired overfitting artifacts that could not be removed by Laplacian regularization without oversmoothing the geometry.

Instead, for our final tracking, we determine the new position of a constrained vertex $\boldsymbol{q}_i^t$, $i \in C^t$, by looking for the scene flow vector $\boldsymbol{d}_i^t$, whose starting position is closest to $\boldsymbol{q}_i^t$. The updated constrained vertex position is then calculated as $\boldsymbol{q}_i^t + \boldsymbol{d}_i^t$. Possible errors introduced by assigning the value of the closest scene flow vector rather than its end point can be compensated for by an optional motion refinement step, which is the subject of Sec. 7.4.2.

Laplacian Regularization For a natural shape-preserving deformation of the face, we regularize the geometry of the target mesh M_c^{t+1} using the differential coordinates of the template mesh $M_c^{t_0}$ (similar in spirit to Bradley *et al.* (2010)). The differential coordinates of $M_c^{t_0}$ encode the shape characteristics of the template surface and encapsulate information about the specific face that we are tracking. If we used the differential coordinates of the current mesh M_c^t, the original shape of the face would not be preserved and the template structure would eventually be "forgotten". Using $M_c^{t_0}$ as a shape prior instead will avoid drift, while still allowing the capture of the low frequency component of strong facial deformations.

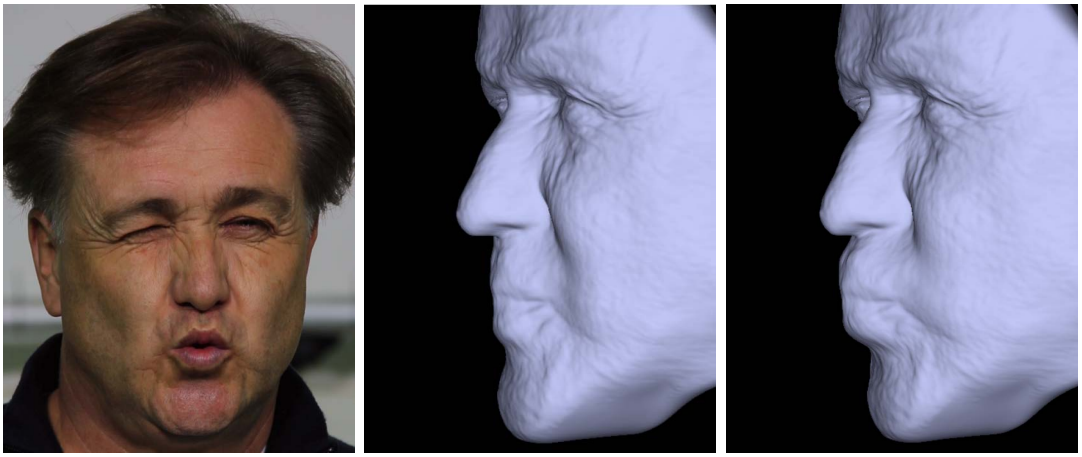


Figure 7.3: Motion refinement. From left to right: input image, corresponding mesh obtained without motion refinement, corresponding mesh obtained with motion refinement. Note how the lips protrude more, as in the input image.

To deform M_c^t to M_c^{t+1} under the influence of the constrained vertices $\boldsymbol{q}_i^t$, $i \in C^t$, we minimize the energy

$$E = \left\| L\boldsymbol{g}^{t+1} - L\boldsymbol{g}^{t_0} \right\|^2 + \mu^2 \sum_{i \in C^t} \left\| \boldsymbol{q}_i^{t+1} - (\boldsymbol{q}_i^t + \boldsymbol{d}_i^t) \right\|^2, \tag{7.2}$$

where L is the Laplacian matrix with cotangent weights of $M_c^{t_0}$ Sorkine (2005). Further, $\boldsymbol{g}^{t+1}$ and $\boldsymbol{g}^{t_0}$ contain the vertex positions of the meshes M_c^{t+1} and $M_c^{t_0}$, and μ is a weighting factor.

7.4.2 Motion Refinement

Two possible sources of error remain in our tracking pipeline: First of all, the Laplacian regularization maintains mesh integrity but may prevent the vertices from moving to their true target positions. Secondly, we can expect a gradual accumulation of motion errors over multiple frames. To compensate for such errors, we introduce a motion refinement step. The idea is to generate a synthetic image pair (I_0^r, I_1^r) by reprojecting the tracked mesh M_c^{t+1} onto the left and right image and to correct its position by minimizing the deviation between (I_0^r, I_1^r) and the ground truth (I_0^{t+1}, I_1^{t+1}). This effectively minimizes the reprojection error. We do this by computing the scene flow between (I_0^r, I_1^r) and (I_0^{t+1}, I_1^{t+1}) and by updating the position of M_c^{t+1} as explained in Sec. 7.4.1.

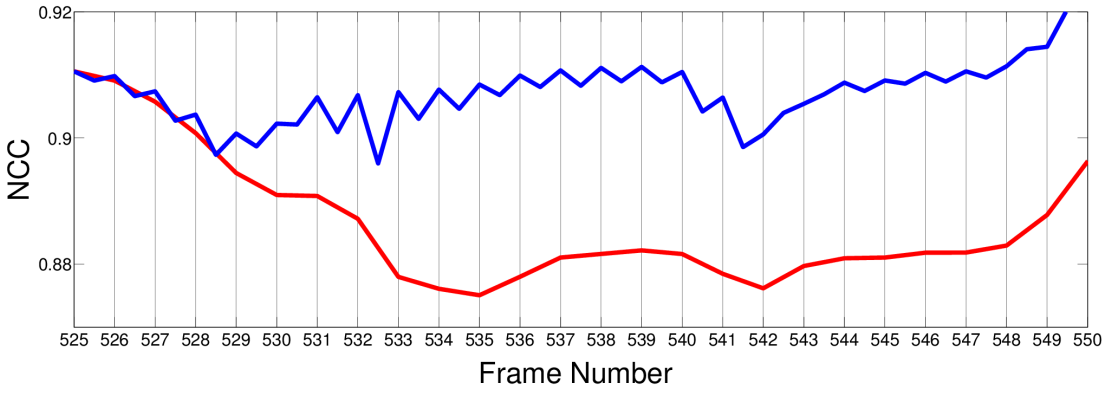


Figure 7.4: Motion refinement. Graph of the NCC for a tracking result with (blue) and without (red) motion refinement.

In Fig. 7.3, we illustrate the effect of motion refinement after capturing 30 frames of the same sequence. We see that the geometry on the right, reconstructed with motion refinement, is closer to what one would expect from the corresponding input image on the left. This visual impression is confirmed quantitatively by a higher normalized cross correlation (NCC) between the reprojected image and the input image. In the bottom row of Fig. 7.4, we plotted the NCC between the reprojected image and the corresponding input image for 25 frames of a similar sequence. For tracking with motion refinement (blue curve), we see that the NCC is consistently higher than it is without it (red curve). From our experience, motion refinement is important for the realistic capture of expressive facial motion, such as the examples in the figure. For such cases, we found that one refinement step per time instance constituted a good compromise between accuracy and computational complexity. This strategy is illustrated in the blue curve by an increase in NNC between each pair of consecutive frames. For less expressive motion such as speech, we found no large improvements in the estimated geometry. For such cases motion refinement could be applied less frequently or even considered optional. Motion refinement assumes that the texture of the face mesh does not change much over time and is thus less effective in the case of changing illumination and cast shadows.

7.5 Shape Refinement

In this section, we explain how we employ shading cues to infer the high-frequency facial geometry detail and add it to the coarse tracked template. The idea here is similar to that proposed in Chapter 5, i.e., employing inverse rendering for high frequency detail estimation. Then, similarly, the shading-based refinement algorithm consists of two steps: First, the lighting and albedo for each frame are estimated, after which both are used to optimize the geometry based on the shading information in the images. In spite of its similarity to that in Chapter 5, the method here uses an albedo clustering that is better adapted to human faces as well as an improved refinement step which yields better results and faster convergence.

7.5.1 Albedo Clustering

In Chapter 5, the surface albedo is assumed to be piecewise uniform with larger coherent regions of similar reflectance. This could be efficiently segmented using a graph-based segmentation method Felzenszwalb & Huttenlocher (2004). However, when recording human faces from nearby camera positions, this assumption is less appropriate. While it is still fair to assume that there are a few albedo groups of vertices, their locations may not be spatially coherent, e.g. due to skin pigmentation, beards, shadows, etc. In contrast to Chapter 5, we use a K-means clustering method to obtain k albedo groups, where vertices of the same group share the same albedo value. Particularly, given a set of initial per-vertex color albedo values $(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n)$, we aim at partitioning the n vertices of the mesh into k groups $S = \{S_1, S_2, \cdots, S_k\}$ to minimize the within-cluster sum of squares:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{\boldsymbol{j} \in S_i} \|\boldsymbol{a}_j - \boldsymbol{\rho}_i\|^2 \quad , \tag{7.3}$$

where $\boldsymbol{\rho}_i$ is the mean of the initial albedo of the vertices belonging to group i. The initial albedo value $\boldsymbol{a}_i$ is calculated from the shading equation with the geometry and lighting provided by the previous time frame. See Chapter 5 for the initial albedo inference. Once the albedo clusters are obtained, we use the same strategy as in Chapter 5 to estimate the incident illumination and the albedo value for each albedo cluster.

An example of our improved albedo clustering strategy is shown in Fig. 7.5, where the different clusters are color coded.

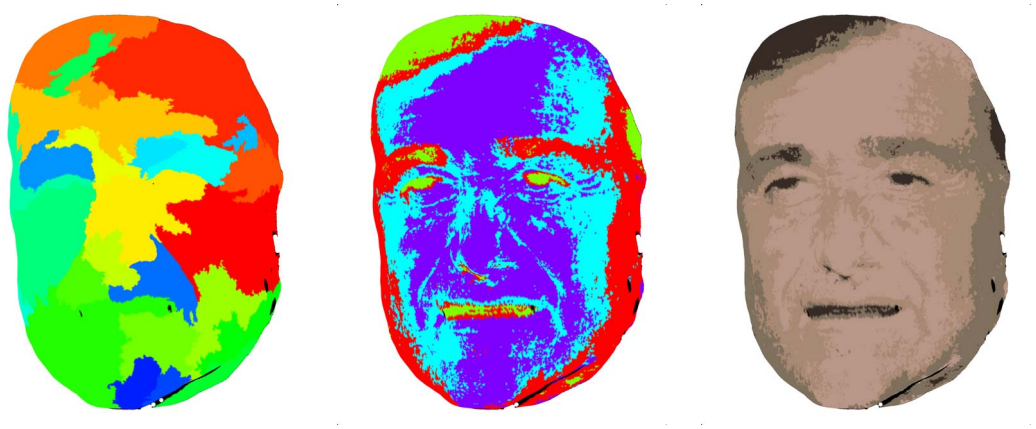


Figure 7.5: Albedo clustering. Left to right: Original spatially coherent clustering from Chapter 5, our new clustering result that corresponds better to typical facial feature distributions (e.g. around eyebrows, eyes etc.), average per material albedo coloring.

7.5.2 Surface Refinement

With the estimated illumination and albedos fixed, the coarse geometry of each frame is refined based on the shading cues in the images. The refined geometry is represented as the displacement of each vertex along its normal direction and is estimated by solving a spatio-temporal MAP inference problem.

Shading Energy Chapter 5 minimizes a cost function that consists of a shading error term (data term) and a prior term (similarity term). Considering the fact that the reflectance of the face will not be purely Lambertian, such refinement will lead to noisy shape details when there are highlights on the skin. To account for this, we add to the energy a second prior term which requires the shape of the face to be spatially smooth (smoothness term). The cost function that we minimize then takes on the form:

$$E = \underbrace{E_{\mathrm{D}}}_{\text{data}} + \underbrace{\lambda_{\mathrm{M}} E_{\mathrm{M}}}_{\text{similarity}} + \underbrace{\lambda_{\mathrm{S}} E_{\mathrm{S}}}_{\text{smoothness}} \quad , \tag{7.4}$$

where λ_{M} and λ_{S} are weighting factors. The data term E_{D} is the shading error that measures the similarity of the shading gradients in the input images I_0^t and I_1^t to the predicted shading gradients:

$$E_{\mathrm{D}} = \sum_i \sum_{j \in N(i)} \sum_{c \in Q(i,j)} \left(r_c(i,j) - s(i,j)\right)^2 \quad , \tag{7.5}$$

where i and j are triangle indices, $N(i)$ is the set of neighboring triangles of triangle i, c is the camera index, $Q(i,j)$ is the set of cameras which see triangles i and j, and $r(i,j)$ and $s(i,j)$ are the measured image gradient and predicted shading gradient. The similarity term E_{M} is a prior term, based on the previous frame geometry, that requires the current refined geometry M_r^t to be similar to the refined geometry of the previous time step M_r^{t-1}, transplanted on the coarse mesh M_c^t. This prior constrains the reconstructed high-frequency shape detail in the face, such as fine folds and laugh lines, to change in a spatio-temporally coherent way. In particular, it takes on the form:

$$E_{\mathrm{M}} = \sum_{i}^{n} \sum_{u,v} \left(\hat{\boldsymbol{n}}_i^t \cdot (\boldsymbol{q}_u^t - \boldsymbol{q}_v^t) \right)^2 \quad , \tag{7.6}$$

where vertices $\boldsymbol{q}_u^t$, $\boldsymbol{q}_v^t$ and $\boldsymbol{q}_i^t$ belong to the same mesh triangle and $\hat{\boldsymbol{n}}_i^t$ is the propagated surface normal based on the already reconstructed high-frequency normal field of the previous time $t-1$. For more details on the propagation of the surface normals, please refer to Chapter 5. In contrast to Chapter 5, however, we define the surface normals in the data and similarity term on triangles. The third, newly-added term in our energy equation (7.4) is the smoothness term E_{S}, which has the following form:

$$E_{\mathrm{S}} = \sum_{i}^{n} \left\| \sum_{j \in N(i)} w_{ij} \left(\boldsymbol{q}_i^t - \boldsymbol{q}_j^t \right) \right\|_2^2 \quad . \tag{7.7}$$

Here $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are the positions of the vertices i and j in the mesh M_r^t, $N(i)$ is the 1-ring neighborhood of vertex i, and w_{ij} are the common cotangent weights Sorkine (2005).

Fast Iterative Minimization The shading energy (7.4) is usually non-linear and not trivial to minimize. In Chapter 5, we employ a patch-based non-linear optimization strategy to refine the geometry within separate vertex patches. Such a strategy, however, imposes a trade-off between run time and quality: While a small patch size may not constrain the neighboring vertices enough to achieve high quality large-displacement shape refinement, a large patch size will take much longer to compute.

Here, we reduce this trade-off by replacing the non-linear optimization of the energy with an iterative linear one. To this end, we replace the only non-linear

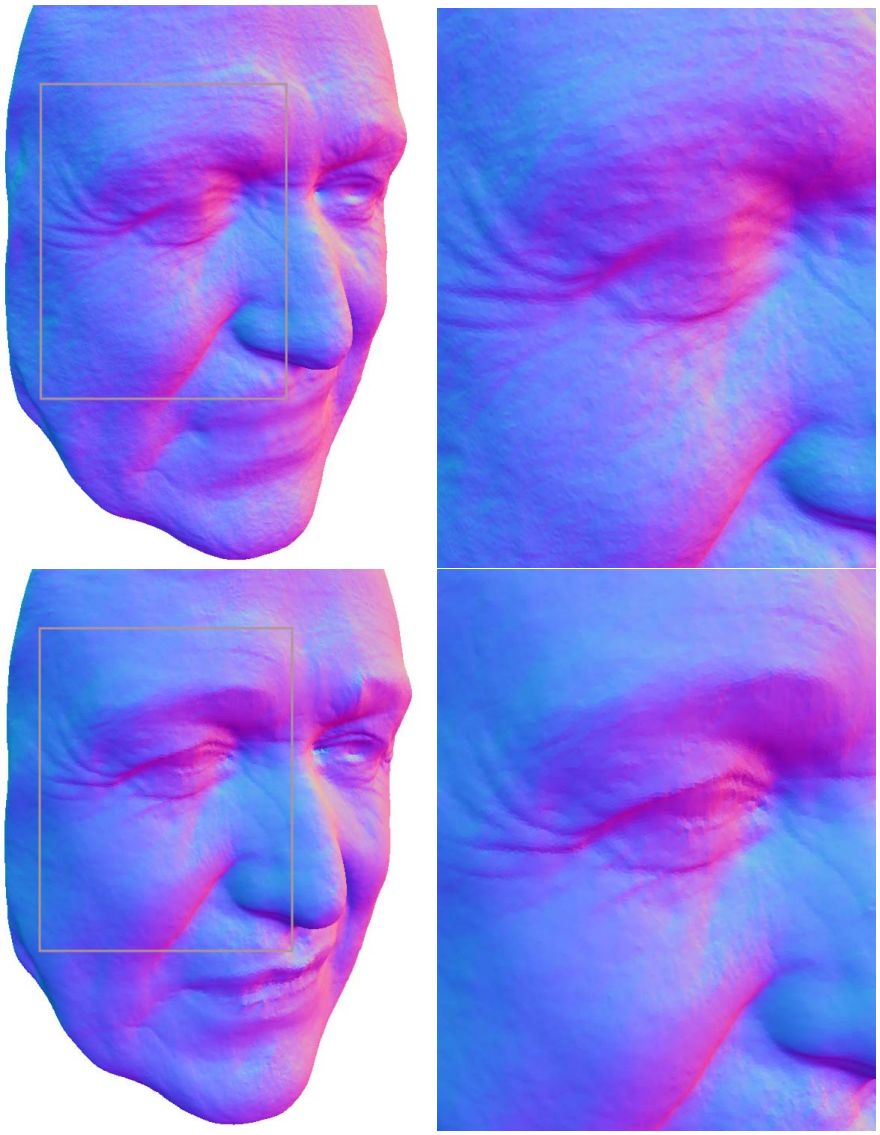


Figure 7.6: Novel Shape Refinement. Top row: results obtained using Chapter 5. Bottom row: results obtained using our current method. Both meshes are colored by normal orientation. The zoom-in shows that we obtain a smoother result with an even higher level of detail(best visible in electronic version).

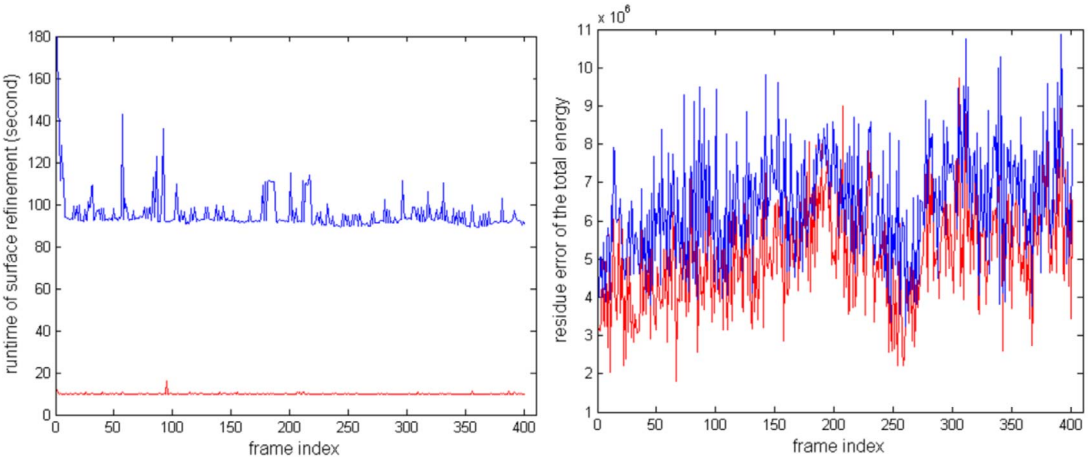


Figure 7.7: Fast Iterative Minimization. Left: run time per frame for a sequence refinement with constant parameters. Right: shading energy per frame. Red: our optimization using the present method, blue: optimization of Chapter 5.

part in the energy (7.4) — the argument of the shading error term — with its first-order Taylor approximation. This way, all terms in the energy become squared linear with respect to the vertex displacement which we aim to optimize. Since each vertex in the resulting energy only has a relation with its direct neighbors, the vertex displacements can be easily found by solving a sparse linear system. A first-order Taylor approximation is only valid when the displacements are small, so in practice we update the vertex positions using the obtained solution scaled by an adjustable step size. We repeat this procedure several times, such that the sequence of newly-defined energies better approximates the original one. In our experiments, we use a step size of 0.7 and iterate 4 times to obtain the final refinement.

Fig. 7.6 shows that the novel shading energy and the iterative minimization strategy lead to superior refinement results compared to those of Chapter 5: our estimated face surface suffers less from noisy artifacts, while exhibiting an even higher level of fine-scale detail such as wrinkles.

Because we can make use of fast sparse linear solvers, e.g., Cholesky decomposition, and because all vertices are optimized simultaneously in each iteration step, we achieve a general speed-up over patch-based non-linear optimization. This is depicted in Fig. 7.7, where the graph shows that our novel iterative minimization strategy reduces computation time by an order of magnitude compared to the non-linear patch-based optimization in Chapter 5 for the same sequence

Figure 7.8: The setups used in our experiments. From left to right: A pair of Canon EOS 550D cameras; the GoPro 3D Hero system.

with constant parameters. The figure also shows that the proposed shape optimization strategy converges to a lower energy, and thus to a better optimal shape, for most frames.

7.5.3 Temporal Postprocessing

After the final shape refinement, there might remain a slight temporal flicker in the visualization of the results due to small differences in the direction of the surface normals. To reduce this effect, we update the normals in the whole sequence by averaging them over a temporal window of size 5 and then adapting the geometry to the updated normals using the method of Nehab *et al.* (2005).

7.6 Results

We evaluate the performance of our approach on real world data captured from three different test subjects with two different stereo camera setups: (I) a pair of Canon SLR cameras in an indoor environment and (II) a pair of GoPro helmet cameras, both indoors and outdoors. In total, five sequences of lengths varying from 300 to 560 frames (12s to 22s) will be presented.

Canon Setup Our first setup consists of two Canon EOS 550D cameras in an indoor environment, as depicted in Fig. 7.8. These cameras record HD video with a resolution of 1920×1088 at 25 frames per second. They are not hardware synchronized and were just started at the same moment, and synchronization is verified by event-based temporal alignment. The green screen in the figure is not a requirement, just a standard feature of the room we used.

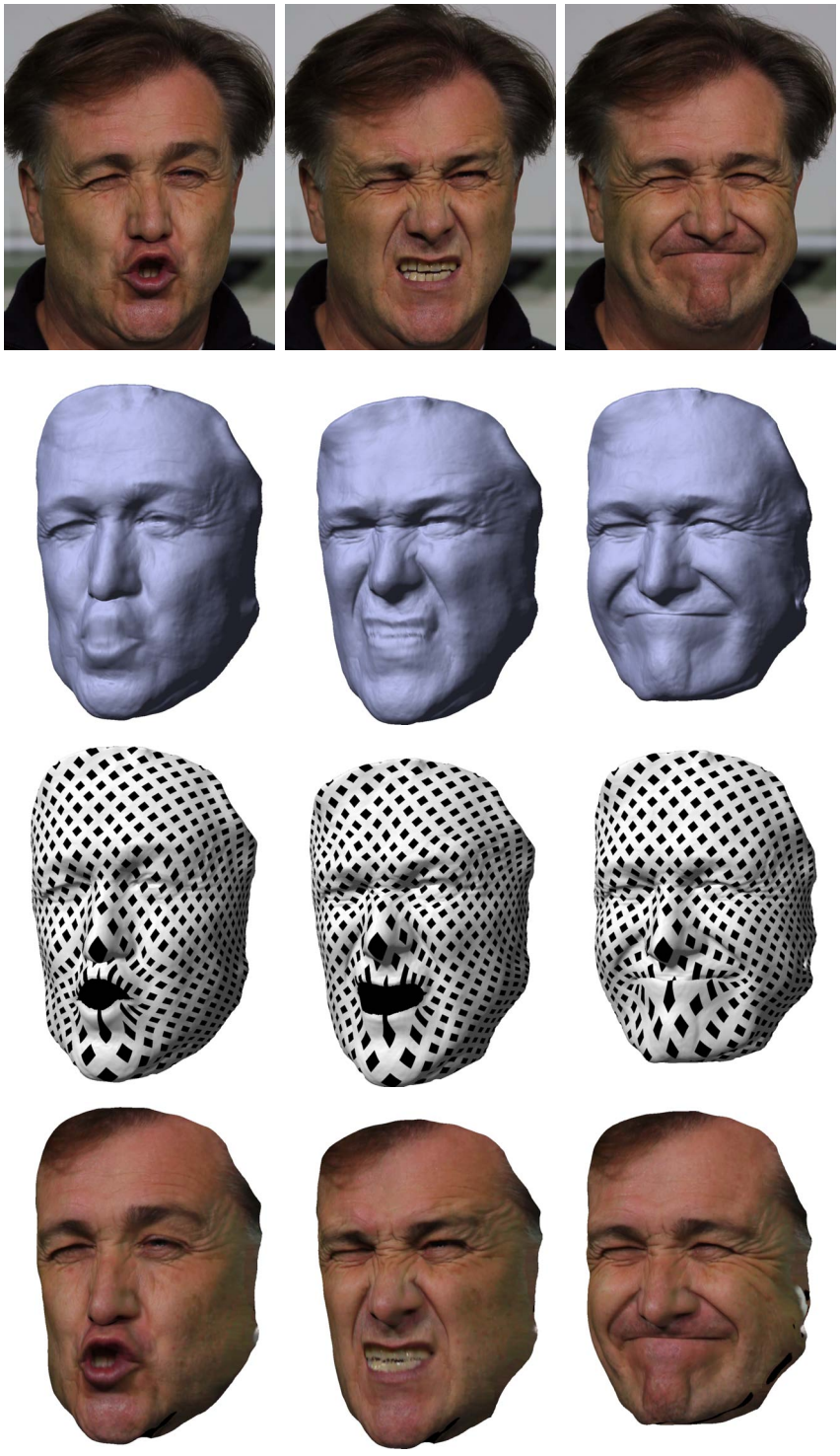


Figure 7.9: Results for a pair of Canon cameras. From top to bottom: the left input image, the corresponding reconstructed mesh, the mesh overlaid with a checkerboard pattern to demonstrate geometric coherence, the mesh colored using projective texturing.

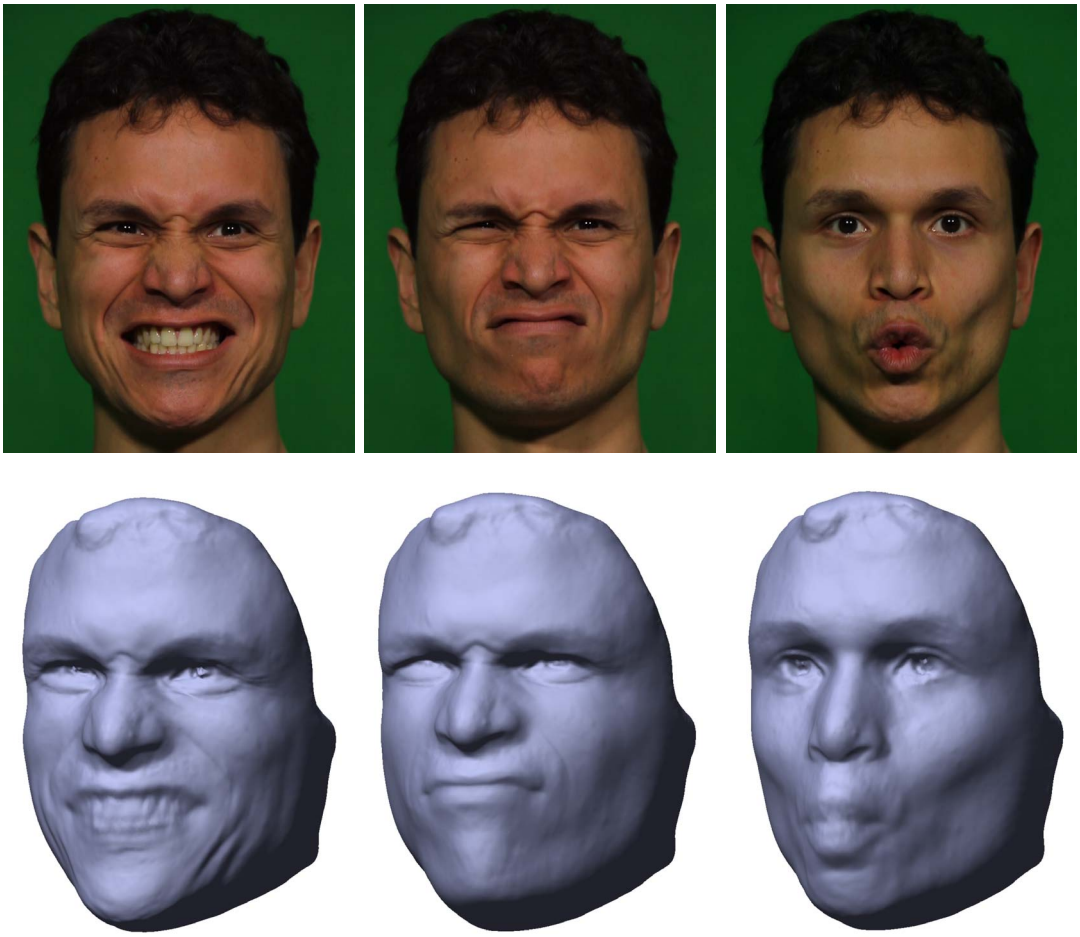


Figure 7.10: Results for a pair of Canon cameras. From top to bottom: the left input image, the corresponding reconstructed mesh.

In Fig. 7.9 we show the results for a subject captured with this setup. All meshes consist of the same set of vertices and have been produced by tracking a single template mesh throughout a sequence of around 300 frames. The number of reconstructed vertices is 100,000. These results illustrate that we are able to capture very expressive facial motion with a level of detail that rivals more complex capture methods using more cameras and controlled lighting. Reconstructions are space-time coherent with no perceivable drift, as illustrated by the checkerboard result. With such high-quality reconstructions, realistic-looking textured faces can be also created via projective texturing with no perceivable ghosting.

In Fig. 7.10 we provide a long captured sequence for a different actor performing both extreme facial gestures and normal conversation. Both types of motion

are captured by our method with high quality. Motion blur, due to the fast movement, makes the sequences especially challenging. However, our approach is robust to this and captures fast motion reliably. Even after 560 frames, our method has hardly introduced any temporal drift.

The parameters used for both experiments are: $\alpha_1 = 5$, $\alpha_2 = 5$, $\beta_1 = 200$, $\beta_2 = 150$, $\beta_3 = 200$, $\rho = 3$, $\lambda_s = 0.1$, $k = 4$, $\lambda_{\mathrm{M}} = 2500$ and $\lambda_{\mathrm{S}} = 10000$. The weight μ was chosen 1 and 0.9 respectively. The run time for sequential, non-optimized code is around 9 minutes per frame on an Intel Xeon@3.1GHz.

GoPro Setup Our second setup uses a pair of GoPro HD Hero cameras that are hardware synchronized and combined in a single housing (Fig. 7.8). The pair records at 1920×1080 and 30 frames per second each. The camera is designed to be used outdoors on bike helmets and is at best comparable to an upscale webcam. Data are challenging due to the smaller baseline, cheap plastic wide angle lenses, the generally higher noise level, the rolling shutter, and potential automatic white balancing which cannot be controlled.

Recordings are captured with the hand-held GoPro HD Hero stereo system in both indoor and outdoor environments. One such setup is depicted in the first row of Fig. 7.11, where a speaking actor is recorded indoors. The general uncontrolled lighting makes this scenario extremely difficult for any facial motion capture algorithm. Moreover, compared to the Canon setup, the face of the actor only makes up a small portion of the HD images. Despite these challenges, we obtain reconstructions which exhibit a fair amount of detail (Fig. 7.11). We are able to capture the face, including the head motion, over extended periods of time with only a little drift.

A second scenario is shown in the second row of Fig. 7.11, where an actor records himself outdoors in bright direct sunlight. For this sequence, the face was captured over 400 frames with equally high quality as in the previous recording. There is a strong shadow from the nose that floats freely over the mouth, and motion refinement in the tracking step treats this incorrectly as physical motion. Although motion refinement was disabled for this sequence, we are still able to achieve very expressive facial motion. The same strong shadow also leads to artifacts on the boundary caused by the shading-based refinement step. These high-frequency effects can be partly alleviated by the use of higher order spherical harmonics (see Chapter 4) to better approximate the visibility and shadow boundaries. This would, however, increase the run time substantially. An option for handling this case is the explicit detection of strong shadows to disable shape

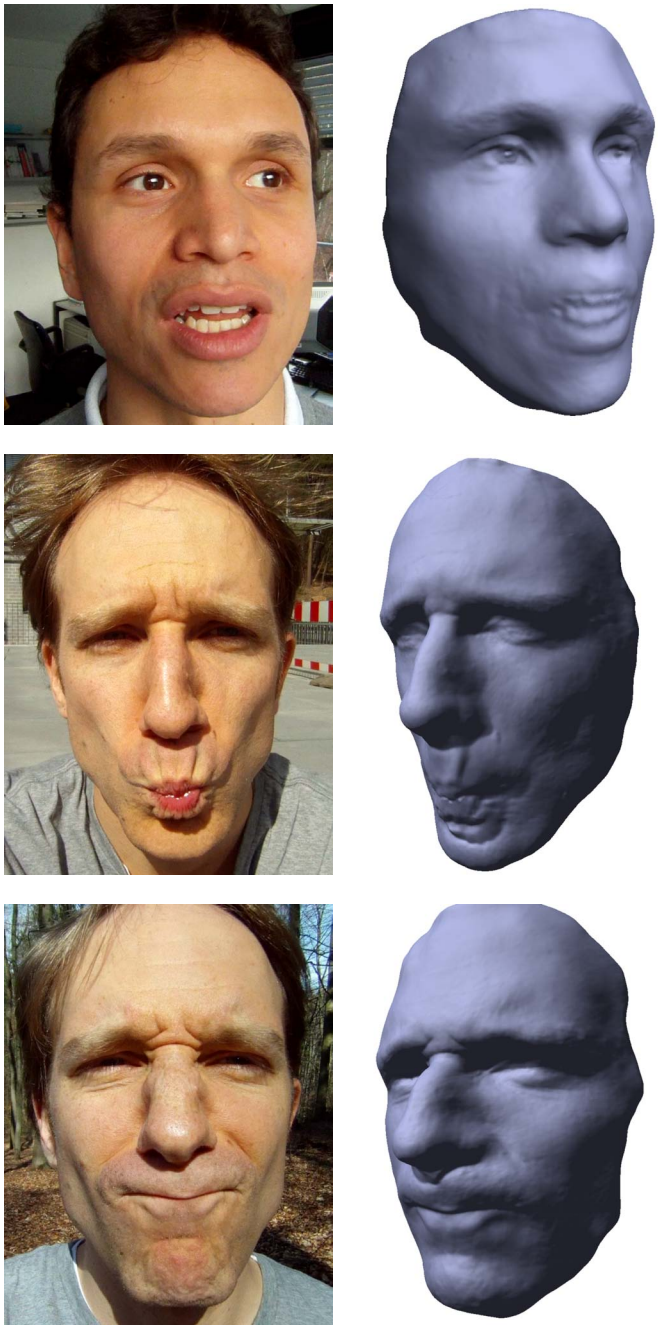


Figure 7.11: Results for a pair of GoPro HD helmet cameras for a person being recorded in an uncontrolled indoor environment (first row of figures), a person recording himself outdoors in bright sunlight (second row of figures) and under changing illumination (last row of figures).

refinement in these regions. We detect shadows by high shading errors and estimate them iteratively with the lighting environment map. The results that we obtain do not have shadow boundary artifacts, but do exhibit less detail. This trade-off between fine detail and shadow artifacts should be chosen with respect to the application in mind.

The last row of Fig. 7.11 shows a third scenario, where an actor records himself while walking in an outdoor wooded setting. This is a very challenging scenario, not least due to the additional background motion and the changing illumination on the face caused by shadows from the trees. Nevertheless, we are able to capture highly detailed and realistic facial motion, which shows that both our tracking and shape refinement pipeline are robust with respect to the aforementioned difficulties.

The parameters used for these experiments are the same as for the Canon sequences, with the exception of $\beta_1 = 300$, $\beta_2 = 200$, $\beta_3 = 300$ and $\lambda_S = 40000$. The weight μ is 0.9 for the indoor sequence and 0.4 for the outdoor sequences.

Discussion Our approach is the first to capture highly detailed facial performances from a single stereo rig under general illumination, and it shows that on-set capture with consumer grade hardware is feasible. Our approach is nevertheless subject to some limitations. One of them arises from strong shadows on the face. Strong moving shadows are a major challenge for the tracking step, which interprets the sharp, moving boundaries as surface motion. We will investigate better ways of detecting shadows and the use of photometric invariants for scene flow computation Zimmer *et al.* (2011) in future work. In addition, temporal drift is reduced by our method, but cannot be completely prevented over extended periods of time. In this context, a combination of our template tracking approach with a key-frame-based regularization Beeler *et al.* (2011) may be promising. Also, scene flow estimation could be further improved by using the refined geometry model as an explicit regularizer rather than relying merely on image constraints. In the future, we will also look into extracting more advanced reflectance and lighting models from the data and to investigate whether this enables further improvements of the results.

7.7 Conclusion

In this chapter, we presented an algorithm to capture high-quality facial performances from a single stereo pair of video streams that were captured in general

environments, even outdoors. This becomes possible through the combination of mesh tracking by the scene flow constraint and a shading-based refinement approach that captures space-time coherent, highly detailed geometry. With our approach, we are able to produce results of a high quality that could not previously be achieved using just a single stereo rig. Our method can make hand-held facial performance capture feasible for everyone. It also opens the door for new applications in on-set facial performance capture, movie postprocessing, etc.

Chapter 8

On-set Performance Capture with a Stereo Camera

8.1 Introduction

Marker-less full-body performance capture methods enable the reconstruction of detailed motion, dynamic geometry, and the appearance of motion actors from multiple video recordings. However, most existing marker-less methods require studios with controlled lighting, a controlled background, and a large number of cameras. These constraints limit the extent to which the method can be applied to many practical feature film productions, as they may require methods that can work on a normal set or on location rather than a separate green-screen controlled stage. If such a method, with the ability to capture detailed moving 3D models of actors on the actual production set, can be developed, it would broadly benefit movie and VFX production.

In previous chapters, we showed how to investigate inverse rendering to relax the constraints of marker-less performance capture on lighting and background, albeit with a multi-view camera setup. While these methods are able to capture detailed models of actors in natural motion and natural apparel without markers, the requirement for a multi-view camera setup is still a big obstacle in the path to on-set performance capture. This is because on a real production set, it is difficult to effectively place the satellite cameras for capturing the motion or geometry, as the environment and scene conditions are very general and chosen with the visual quality of the shot in mind. This is usually orthogonal to the requirements that vision-based tracking algorithms have for robust operation. While a performance capture method that can succeed with the cameras placed within a limited space

on set is always desired, the question of how to choose an appropriate number of cameras and appropriately place them on a production set is still open.

Currently, performances of real actors in a scene are frequently composited with virtual renditions of actors during post-processing. One example is the movie *Pirates of the Caribbean*, where real actors in a scene wear marker suits. The Imocap system is used to track their skeletal motion from the primary camera and a few satellite cameras and, in post-production, the actors in marker suits are replaced with virtual renditions. This common example shows the importance and tremendous difficulty of the task, since even the skeletal tracking alone required substantial manual marker labeling by an operator. If we could capture detailed motion *and* surface geometry automatically under the more general lighting conditions and backgrounds of a production set, while using only production stereo cameras, then actors would benefit by being able to work on the real set while being captured, more realistic overlays of virtual actors could be created, more detailed pre-visualizations of CG augmented actors on set could be created, and the recovery of a 3D model underlying each actor in the scene would enable novel editing possibilities such as appearance modifications.

In this chapter, we describe a novel performance capture algorithm which works towards this goal. It enables us to capture the full body skeletal motion and detailed surface geometry of one or more actors using only a single stereo pair of video cameras. It is designed to work without additional depth sensors, such as Kinect, which only work indoors, have a limited range and accuracy, are not part of standard production cameras, and may interfere with other on-set equipment. The low-baseline stereo camera rig is permitted to move during recording. This setting is akin to modern movie production sets with a primary stereo camera. Our algorithm succeeds under uncontrolled lighting, non-frontal body poses of the actors, and scenes in which actors wear general apparel with non-Lambertian reflectance. It also succeeds in front of general scene backdrops where classical background subtraction would be infeasible. The work presented here was published in Wu *et al.* (2013).

8.2 Method Overview

Our algorithm tracks and deforms a template model for each actor in the scene such that it optimally aligns with stereo input images. An overview is shown in Fig. 8.1. Input to our algorithm is a stereo video sequence of a scene filmed with a camera rig that can freely move, as well as a light probe image of the set without

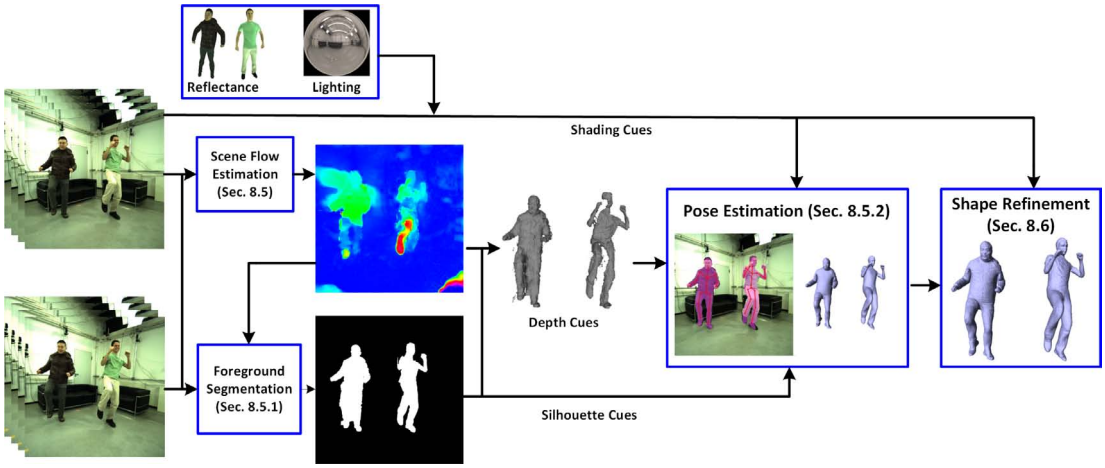


Figure 8.1: Overview of our performance capture method.

actors. We also expect, for each actor, a static triangle mesh shape template with an embedded kinematic skeleton that can be obtained from a laser scan or from image-based reconstruction. Instead of relying on simple light transport assumptions, and assuming Lambertian surface reflectance as in previous chapters, our performance capture method applies knowledge about the incident illumination and a detailed spatially-varying BRDF of every actor in a scene for both skeletal pose estimation and for reconstruction of detailed surface geometry. Therefore, we expect as additional input a spatially-varying parametric BRDF surface model for every actor, captured prior to stereo recording. In practical productions, reconstruction of such a reflectance model for each actor is becoming standard and can be performed with a light stage Vlasic *et al.* (2009). However, inspired by the capture methods under general illumination Li *et al.* (2013), we describe in this chapter a lightweight method to estimate the BRDF based on multi-view video footage of a moving actor recorded under standard studio lighting (Sec. 8.4).

Our main contribution is a new skeletal pose estimation approach. It relies on a new stereo-based foreground segmentation algorithm that employs appearance cues, scene flow, pose reconstruction results from previous frames, and stereo coherence to reliably segment out actors in front of general backgrounds (Sec. 8.5.1). Pose estimation is based on minimizing a new energy function that measures the model-to-image consistency based on the segmented silhouettes, the depth map given by scene flow, and the shading consistency based on a full diffuse and specular surface BRDF model (Sec. 8.5.2).

First, our algorithm captures the skeletal pose of each actor together with surface geometry which lacks high frequency shape detail such as cloth folds (Sec. 8.5). Second, this detail is reconstructed by a new inverse rendering approach that refines the coarse geometry using shading-based dynamic scene refinement based on the scene illumination and the full surface BRDF (Sec. 8.6).

We demonstrate the performance of our algorithm on a variety of scenes with general uncontrolled lighting, and scenes showing several actors performing motions with difficult occlusions and out-of-plane motions. We also show results on footage with apparel with challenging non-Lambertian appearance, and scenes filmed with a moving camera rig. We qualitatively and quantitatively demonstrate the accuracy of our method and the importance of each step, and show that the quality of our reconstructions enables appearance editing of actors in video (Sec. 8.7).

8.3 Image Formation Model

In this chapter, we are aiming to reconstruct the scenes with a more general and more expressive BRDF, which means in addition to diffuse reflectance our image formation model here also needs to represent the non-Lambertian or specular reflectance. As introduced in Chapter 2, the general BRDF can be represented by a combination of diffuse reflectance and specular reflectance. In detail, here we represent the diffuse component as Lambertian albedo, and the specular component using a simplified Torrance-Sparrow model Torrance & Sparrow (1967). Then, our reflectance model can be written as:

$$\rho\left(\omega_i, \omega_o\right) = k_d + \frac{k_s}{4\pi\sigma_b^2 \cos\theta_i \cos\theta_o} \exp\left(-(\theta_h/\sigma_b)^2\right) \quad , \tag{8.1}$$

where k_d and k_s are the diffuse and specular albedos, θ_i, θ_o, and θ_h are the incoming light direction, the viewing direction, and the half angle, respectively, all defined with respect to the surface normal, and σ_b is the surface roughness. As discussed in Chapter 2, the reflection equation for general BRDF then can be represented as:

$$B(\alpha, \beta) = k_d B_d(\alpha, \beta) + k_s B_s(\alpha, \beta) \quad , \tag{8.2}$$

where B_d and B_s are respectively the reflected irradiance from the diffuse component and the specular component, and (α, β) are the spherical coordinates of

the surface normal. For diffuse irradiance, B_d can be efficiently represented using a low order SH, which has the form:

$$B_d(\alpha,\beta) = \sum_{l=0}^{N_D} \sum_{m=-l}^{l} \Lambda_l \, L_{lm} \, \hat{\rho}_{dl} \, Y_{pq}(\alpha,\beta) \ , \tag{8.3}$$

where $\hat{\rho}_{dl}$ are the SH coefficients for the clamped cosine function, Λ_l is the normalization constant, and N_D is the order of SH, which is taken to be $N_D = 4$, similar to the previous chapters. The specular irradiance B_s can also be represented using SH, which takes the form

$$B_s(\alpha,\beta) = \sum_{l=0}^{N_S} \sum_{m=-l}^{l} \Lambda_l \, L_{lm} \, \hat{\rho}_{sl} \, Y_{pq}(\alpha',\beta') \ , \tag{8.4}$$

where $\hat{\rho}_{sl}$ are the SH coefficients of the properly reparameterized BRDF, (α',β') is the reparameterized spherical angle of (α,β) with respect to the central direction of BRDF, and N_S is the order of SH, which is generally higher than for the Lambertian case due to the high-frequency component in the specular reflectance function. In this chapter, we take $N_S = 10$ and will reduce it accordingly if specular BRDF parameters are obtained.

Based on the image irradiance equation described above, we first describe how to inversely estimate the BRDF function from a multi-view and multi-lighting image sequence in Sec. 8.4. Afterwards, an analysis-through-synthesis pose estimation method is explained in Sec. 8.5, and a new shape refinement method based on Eq. (8.2) is introduced in Sec. 8.6.

8.4 Template and Reflectance Reconstruction

A first input to our algorithm is a static triangle mesh shape template M_c for each tracked actor. We use a laser scanner, but M_c could also be obtained via image-based reconstruction. The template is purposefully smoothed to remove static high-frequency shape detail. A skeleton with 20 joints and 37 degrees of freedom controls the motion of the shape template via skinning.

A model of surface reflectance for every actor is a second important prerequisite enabling stable binocular performance capture in general scenes (see Sec. 8.5). Such a model can be captured using a light stage Vlasic *et al.* (2009). However, with wider applicability in mind, we employ an alternative solution that is based on a simpler studio setup and is inspired by methods that are able to capture

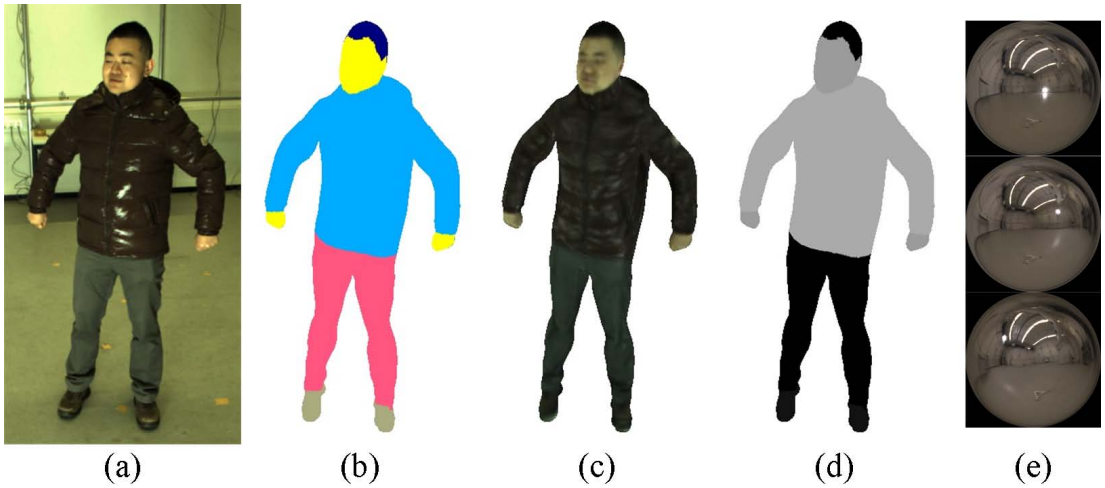


Figure 8.2: Reflectance estimation: (a) input image, (b) material segmentation, (c) estimated spatially varying diffuse albedo, (d) estimated per-segment specular albedo, (e) the light probe images.

the BRDF under general illumination Li *et al.* (2013). Our approach consists of three light sources placed vertically at different heights and a calibrated multi-view camera system (see Fig. 8.2), a setup that is close in spirit to Theobalt *et al.* (2007). The actor is recorded performing a simple rotational motion with all three light sources turned on sequentially. Prior to recording, a ground truth environment map is captured for each such lighting condition and projected into spherical harmonics space. Then, we use the performance capture algorithm proposed in Chapter 6 to track this simple sequence with our shape template.

The rotating motion of the performer allows us to collect reflectance samples of visible surface locations in the camera views. These samples are captured under different illumination and viewing conditions covering a range of azimuthal angles, while the vertical displacement of the light sources gives us measurements on different elevation angles. From these samples, we estimate the BRDF parameters k_d, k_s and σ_b (see Sec. 8.3). Although it is desirable to estimate these parameters for each point on the template mesh, the high-frequency reflectance component is particularly hard to estimate from a relatively sparse set of samples. Similar to the previous chapters, to make the calculation tractable, we assume the surface to comprise a discrete set of b materials K_b (such as skin), each with a constant specular reflectance. This discrete set of materials is manually segmented in the first frame (see Fig. 8.2b), but could also be found via color clustering. Another simplifying observation is that many common materials are dielectric, i.e., the

generated highlights are of the same color as the light source. Following this assumption, we can represent the specular albedo k_s as a scalar value. Thus, we solve for a per-vertex k_d, a per-patch specular albedo k_s, and a per-patch surface roughness parameter σ_b.

BRDF estimation is performed in an iterative coarse-to-fine way. In a first iteration, we assume that all BRDF parameters are constant for all vertices of a material K_b. Then, we minimize the error between the rendered model under the calibrated lighting and the input frames:

$$E_K^b = \sum_f \sum_{i \in K_b, c \in N_c} w_{\boldsymbol{q}_i,c} \, \| k_d B_d + k_s B_s(\sigma_b) - I_c(\boldsymbol{q}_i, f) \|, \tag{8.5}$$

where f is the frame index, i is the vertex index, c is the camera index, and $w_{\boldsymbol{q}_i,c}$ is a weighting factor. The surface normals of the coarse model reported by performance capture are too coarse to estimate the reflectance reliably. Therefore, we interleave a refinement of the surface normal orientations using a shape-from-shading approach similar to Zhang *et al.* (1999) with the estimation of the reflectance. We perform normal refinement for each camera view. We iterate normal refinement and reflectance estimation, typically twice. After the first iteration of BRDF and normal estimation, we allow the diffuse albedo k_d to vary for every vertex, while keeping k_s and σ_b fixed per material. To prevent k_d and k_s from being negative in the optimization, we reparameterize them as $k_d{=}r_d^2$ and $k_s{=}r_s^2$, and optimize r_d and r_s instead. All optimization steps are performed with a conjugate gradient solver. We start by setting $N_S{=}10$, and when the BRDF parameters are obtained, we adaptively reduce N_S for each material segment using a strategy similar to Ramamoorthi & Hanrahan (2002) to reduce processing time.

8.5 Skeletal Motion Estimation

It is our goal to estimate detailed surface and skeleton motion of actors in general clothing, who perform general motion in sets with no controlled background, merely from the video footage of a possibly moving stereo camera rig. Compared to previous multi-view performance capture algorithms that operate with tens of cameras and in front of a green screen for easier background subtraction, the drastically reduced set of views and the uncontrolled environment represent a previously unseen challenge. Thus, we need to fundamentally rethink which data

cues to use for tracking, how to measure the model-to-image data consistency, and how to optimize the pose and shape parameters of the template model.

To meet this challenge, our method is the first to jointly employ shading cues from a full BRDF model, depth information, and motion information extracted from binocular views, and also to robustly extract foreground regions representing actors, all from binocular footage in general scenes. First, a light probe image of the empty set is captured, assuming that the lighting is constant for the duration of the recording. Then, we employ a variational approach similar to Valgaerts *et al.* (2010) to compute the 3D scene flow between each consecutive pair of frames (see Chapter 2 for how to estimate the scene flow). This approach computes optical flows in each camera view and 3D stereo geometry for each time step, both of which are used by our algorithm.

Performance capture now subsequently processes pairs of stereo video frames, by alternating the following two steps:

1. A segmentation method is applied to segment out the regions in the depth maps corresponding to persons in the foreground, even if the stereo rig is moving and the background has a general appearance and shape (Sec. 8.5.1). To succeed in this challenging setting, the segmentation method jointly relies on color information, a scene flow-induced body shape prior derived from previous body poses, and stereo constraints between input image pairs. Segmentation produces a depth region of the person to be tracked, whose outlines provide additional silhouette cues for performance capture.

2. The current pose and shape of the actor are found by optimizing a pose error (Sec. 8.5.2). To this end, we employ a tracking algorithm that, for the first time, jointly relies on appearance cues from a full BRDF with diffuse and specular component, silhouette cues, and scene flow information.

8.5.1 Foreground Segmentation

Automatically obtaining clean segmented regions of depth belonging to persons in the foreground is a prerequisite for reliable binocular full body performance capture. Many previous segmentation approaches used color alone for segmenting foreground objects in video. Unfortunately, the colors of foreground objects in general scenes can be very similar, leading to segmentation errors. Often only manual intervention can resolve these problems Rother *et al.* (2004). However, even for multi-view performance capture of interacting persons in front of a green

screen, color information alone was found to be insufficient for labeling persons in video Liu *et al.* (2011). Depth thresholding alone is also not a reliable cue to segment out the person in a scene since, depending on the surrounding geometry, the person may not be the closest object to the camera. Finally, depth or image differencing alone is not suitable, since with a moving camera rig the background model would need to be permanently updated and possibly tracked with a structure-from-motion approach, which is error-prone with a dynamic foreground.

To succeed with a sparse set of binocular views, a general background, and a possibly moving rig, we employ a Markov Random Field (MRF)-based segmentation approach that combines evidence from a variety of scene cues to obtain a reliable segmentation of the persons in the foreground in both input views, and thus in the stereo depth. Foreground segmentation was also employed for motion tracking by Brox *et al.* (2006, 2010) in a multi-view setting by combining appearance cues, modeled by a Gaussian distribution, with a shape prior, provided by the object contour at the current pose. They evolve the object contour by minimizing a non-linear energy, which is sensitive to local minima. Here, we formulate the segmentation as a labeling problem which can be solved efficiently by a graph cut algorithm, and we model the appearance by a Gaussian mixture model (GMM) which enables the segmentation to work for textured objects. Further, we include a shape prediction by the estimated scene flow to obtain a more accurate shape prior and add a new stereo constraint as a consistency check between both cameras.

For every time step, segmentation is performed in two stages: 1) First, pixels in the left and right images are labeled separately as *person in the foreground* or as *background*. In case of multiple actors in the scene, a separate two-label segmentation is solved for each person. 2) Second, the segmentations of each person from both views are fused.

Stage I: In the first stage, the segmentation finds the lowest energy (maximum likelihood) configuration $L = \{l_p \mid p \in 1, \ldots, N_p; l_p \in \{0, 1\}\}$ of the MRF, assigning binary labels l_p to each of the N_p pixels. The energy is defined as follows:

$$D_1^S(L) = \sum_{p \in P} \lambda_A D_p^A(l_p) + \lambda_G D_p^G(l_p) + \lambda_S D_{pq}(l_p, l_q) \ . \tag{8.6}$$

$D_p^A(l_p)$ is a likelihood term penalizing the assignment of label l_p to pixel p based on its color, $D_p^G(l_p)$ is a shape prior exploiting the fact that the body model pose in the previous frame is known and the scene flow between the previous and the current frame is available, and $D_{pq}(l_p, l_q)$ is a regularizing contrast term which

favors pixels having the same label when their color is similar. The weighting factors are experimentally set to $\lambda_A = 2$, $\lambda_G = 10$ and $\lambda_S = 50$.

Color Likelihood A separate color model is used for foreground ($l = 0$) and background ($l = 1$). For both, it is implemented as a GMM of RGB colors with $K_G = 6$ mixture components. The k-th Gaussian $N\big(I(p) \,|\, \mu_k^l, \Sigma_k^l\big)$ in the GMM corresponding to label l is parameterized by a mean μ_k^l, a covariance Σ_k^l, and a weight ω_k^l. Given the pixel color $I(p)$, the appearance cost $D_p^A(l)$ for assigning to it a label l is defined as the negative log-likelihood of:

$$p\big(I(p) \,|\, \mu_1^l, \Sigma_1^l, \omega_1^l, \ldots, \mu_{K_G}^l, \Sigma_{K_G}^l, \omega_{K_G}^l\big) = \sum_{k=1}^{K_G} \omega_k N\left(I(p) \,|\, \mu_k^l, \Sigma_k^l\right) \quad . \tag{8.7}$$

The GMMs for foreground and background are continuously retrained over time to increase robustness under lighting and appearance changes. To train the GMMs for the current frame, we take the foreground and background regions of the previous frame and warp them to the current frame by means of the optical flow computed as part of the scene flow estimation. The colors of the warped regions are used for training the GMM models of the current frame. Fig. 8.3 (b) and (c) show the results of assigning pixels to the foreground or background using the color term for the input image of Fig. 8.3 (a). As shown, the color term is able to distinguish most of the foreground. However, it may not be sufficient when the foreground color is similar to the background, e.g., the lower foot in Fig. 8.3 (a), which leads to an incorrect segmentation result as seen in Fig. 8.3 (d).

Shape Prior The shape prior measures the label assignment cost based on a prediction of the pose of the model in the current time step, given that its pose in the previous time step is known and motion is smooth. We model this term by warping the previous pose of the shape model onto the current frame via scene flow. The warped model is projected onto the image and we build a heat map H^G based on the pixel's distance to the outer contour of the projected model. The shape prior cost is defined as the negative logarithm of:

$$H_p^G(l_p) = \begin{cases} \frac{1}{1+\exp\left(-d_p^2/(2\sigma_p^2)\right)} & l_p = \hat{l_p} \\ \frac{1}{1+1/\exp\left(-d_p^2/(2\sigma_p^2)\right)} & l_p \neq \hat{l_p} \end{cases} , \tag{8.8}$$

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

Figure 8.3: Foreground segmentation: (a) input image, (b)+(c) color likelihood for background and foreground (white: high, black: low), (d) segmentation result using only color likelihood, (e) color term and shape prior likelihood, (f) final segmentation result using all components.

where d_p is the distance from pixel p to the nearest contour point, $\hat{l_p}$ is the pixel label given by the warped projected model, and σ_p is experimentally set to 5 for all experiments. Fig. 8.3 (e) shows the cost function of assigning pixels to the background, which helps to correctly segment the foot part to the foreground in Fig. 8.3 (f).

Smoothness Term The contrast term D_{pq} takes the same form as described in Liu *et al.* (2011) and is defined as:

$$D_{pq}(l_p, l_q) = \begin{cases} \frac{\gamma}{s(p,q)} \exp\left(\frac{-\|I_p - I_q\|^2}{2\sigma_c^2}\right) & l_q \neq l_q \\ 0 & l_p = l_q \end{cases}, \tag{8.9}$$

where $s(p, q)$ is the spatial distance between the pixels.

The minimum energy (8.6) is found via graph cuts Boykov & Funka-Lea (2006). For efficiency, segmentation is performed for a conservatively extended bounding box around the foreground actor, centered at the location from the previous frame warped by the scene flow. Pixels outside the box are labeled as background.

Stage II: In the second stage, we perform another segmentation of each image by taking into account information from the other camera. Specifically, we augment the MRF energy such that for each pixel in the current view, we check the consistency with the segmentation in the other view. We derive a stereo-based

confidence measure by warping the segmentation of the other view into the current view using scene flow. If the warped segmentation assigns the same label to a pixel in the current view, the pixel is marked as trusted. Then, we retrain the color GMMs for the foreground and background using only trusted pixels in both views. Finally, another graph cut segmentation is performed by minimizing

$$D_2^S(L) = \sum_{p \in P} \lambda_A D_p^A(l_p) + \lambda_G D_p^G(l_p) + \lambda_S D_{pq}(l_p, l_q) + \lambda_O D_p^O(l_p). \tag{8.10}$$

The main extension is the added stereo constraint D_p^O. It assumes the value 1 if trusted pixels are assigned a different label than in Stage I, and 0 otherwise. For untrusted pixels, D_p^O is set to 0.5 for both labels. The weighting factor λ_O is experimentally set to 100.

8.5.2 Pose Estimation

Given a template model of the actor, including a rigged and skinned 3D mesh with reflectance information for each vertex, we track the motion of actors in a binocular input video recorded in an arbitrary uncontrolled environment. As is common in related work, we formulate this as a sequential problem. Given the pose and the geometry M_c^{t-1} at time $t-1$, and two pairs of images at times $t-1$ and t respectively, we want to estimate the skeletal pose at time t. We formulate this as an energy minimization based on the constraints coming from the cues obtained in the previous steps. Gall *et al.* (2009) employ the silhouette and feature constraints for pose estimation in a multi-view setup, which is not enough for our setup (see the comparison in Sec. 8.7). Our energy for pose estimation takes three terms. The first term E^S encodes information from shading cues and measures the difference between the captured images and a rendered version of the character based on the reflectance and the captured environment map. The second term E^G comes from the depth cues, and it measures the difference between our current pose and a depth map of the current image pair calculated as a by-product of the scene flow method. The third term E^H contains the silhouette cues and measures the difference between the projected contour of the mesh at the current pose and the segmented silhouette. The three terms are combined into a single energy term:

$$E^T = \beta_S E^S + \beta_G E^G + \beta_L E^H \ , \tag{8.11}$$

where β_S, β_G, and β_L are weighting factors. We optimize this energy as a function of the skeletal joint angle parameters using a simple conditioned gradient descent

method similar to Stoll *et al.* (2011). The weighting factors are experimentally set to $\beta_S = 1$ and $\beta_L = 10$ for all sequences, while $\beta_G = 20$ for sequences with moving cameras and $\beta_G = 10$ for all other sequences.

Shading Term Similar to previous chapters, the shading energy E^S measures the similarity between a rendered image of the current pose of the actor under the known lighting and reflectance and the captured images. In contrast to previous chapters, we do not assume Lambertian reflectance, but propose one of the first methods to employ a full BRDF model with diffuse and specular reflectance as cues in a 3D pose tracking framework. We demonstrate in the experimental section (Sec. 8.7) that by relying on this more advanced light transport model, we can obtain more accurate and more robust tracking results even with sparse input data captured in general environments. For a single camera c, we write:

$$E_c^S = \frac{1}{N_c^s} \sum_i \left(B\left(c, \boldsymbol{q}_i^t, \boldsymbol{n}_i^t\right) - I_c^t(x_i^t, y_i^t) \right)^2 \quad , \tag{8.12}$$

where N_c^s is the number of visible vertices in camera c, (x_i^t, y_i^t) is the projection of the surface vertex $\boldsymbol{q}_i^t$, $\boldsymbol{n}_i^t$ is the corresponding surface normal, and B is the radiance calculated according to Eq. (8.2). While lighting and reflectance functions are constant, the vertex positions $\boldsymbol{q}_i^t$, the projections (x_i^t, y_i^t), and the normals $\boldsymbol{n}_i^t$ depend on the pose parameters of the model. Taking the same manner with that in Chapter 6, we can calculate analytical derivatives of this function using a Taylor expansion, by ignoring potential visibility changes in the vertices.

Depth Term We estimate per-camera depth maps as part of the scene flow computation. Using the segmentation obtained in Sec. 8.5.1, we remove the background from the depth map. The segmented foreground depth is then refined by removing interpolated depth values at occlusion boundaries via triangle normal orientation thresholding relative to the viewing direction. Based on the filtered foreground depth map, the second component of the pose energy encodes iterative-closest-point-like constraints:

$$E_c^G = \frac{1}{N_c^g} \sum_i \left(\boldsymbol{q}_i^t - c\left(\boldsymbol{q}_i^t\right) \right)^2 \quad , \tag{8.13}$$

where $c\left(\boldsymbol{q}_i^t\right)$ is the corresponding 3D point for vertex $\boldsymbol{q}_i^t$ in the reprojected depth map of camera c based on an approximate nearest neighbor search.

Silhouette Term Following our segmentation, the contour pixels of an actor in the foreground can be conveniently detected in each camera view, enabling us to define a silhouette consistency term. For each of the N_c^h contour pixels, we define a projection ray that can be parameterized as a Plücker line $H_i = (\boldsymbol{s}_i, \boldsymbol{t}_i)$ Gall *et al.* (2009). The silhouette consistency term sums the distance between each line H_i and its closest vertex $\boldsymbol{q}(i)^t$ on the body model:

$$E_c^H = \frac{1}{N_c^h} \sum_i \left(\boldsymbol{q}(i)^t \times \boldsymbol{s}_i - \boldsymbol{t}_i \right)^2 \quad . \tag{8.14}$$

If more than one person is present in the scene, the steps in this section are run for each person separately.

8.6 Shape Refinement

Skeletal tracking yields the coarse shape of each actor in the scene at every time step. However, fine-scale surface detail visible in the images is missing. We recover this with an extended version of the photometric refinement process described in Chapter 5. We formulate this problem as a spatio-temporal MAP inference, where the cost function takes the form:

$$\psi\left(\boldsymbol{g^t}\right) = \phi\left(I^t \,|\, \boldsymbol{g^t}\right) + \phi\left(\boldsymbol{g^t} \,|\, \boldsymbol{g^{t-1}}\right) \ , \tag{8.15}$$

where $\phi\left(I^t \,|\, \boldsymbol{g^t}\right)$ is the shading error that measures the similarity of the image gradients in the input image I^t to the predicted rendered shading gradients according to the image reflectance equation described in Sec. 8.3. The unknown $\boldsymbol{g^t}$ represents the refined surface geometry for every vertex as a displacement from M_c^t in the local normal direction. The term $\phi\left(\boldsymbol{g^t} \,|\, \boldsymbol{g^{t-1}}\right)$ is a prior that requires the current refined surface geometry to be similar to the refined surface geometry of the previous time-step, transformed to the current time-step via skeleton-based deformation and surface skinning using the pose parameters obtained in Sec. 8.5. Please refer to Chapter 5 for details.

Unlike Chapter 5, we adapt the geometry refinement approach to explicitly consider a full diffuse and specular BRDF, rather than just diffuse reflectance. Our method is related to previous stereo methods that phrase multi-view consistency under general surface BRDFs, e.g., Davis *et al.* (2005), but unlike these we do not require images under multiple and often calibrated lighting conditions. Since we are able to exploit the information in the full BRDF, our present method

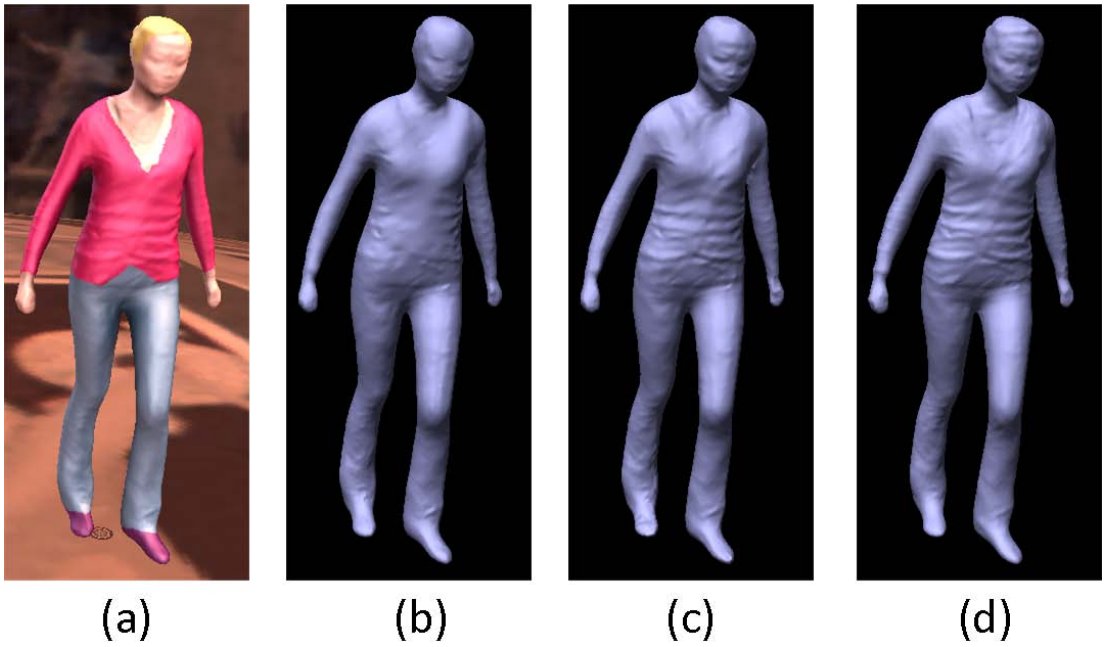


Figure 8.4: Surface refinement on a synthetic test sequence: (a) one of two input images, (b) refined shape using the method in Chapter 5, (c) refined shape using our present method, (d) ground truth shape.

not only works as well as that of Chapter 5 on diffuse surfaces with sparse binocular input data, it also successfully recovers surface detail on very specular surfaces where the previous method would fail, e.g., the specular jacket in the bottom row in Fig. 8.5. As a final result, high frequency shape detail on the surface, such as fine folds and creases, are recovered in a spatio-temporally coherent way. To optimize this energy function, we employ the Levenberg-Marquardt algorithm, which is similar to that used in Chapter 5. Fig. 8.4 shows a comparison of our refinement method with the method described in Chapter 5 on a specular surface.

8.7 Results

We recorded 3 test sequences consisting of over 1300 frames. The data was recorded with a stereo rig with a baseline of $\approx$22 cm at a resolution of 1024×1024 pixels and at a frame rate of 45 fps. Each sequence shows two people wearing casual clothing performing a variety of different motions in front of a general background. The scenes provide various challenges, such as moving cameras, specular apparel, close contact with background objects, and partial occlusions

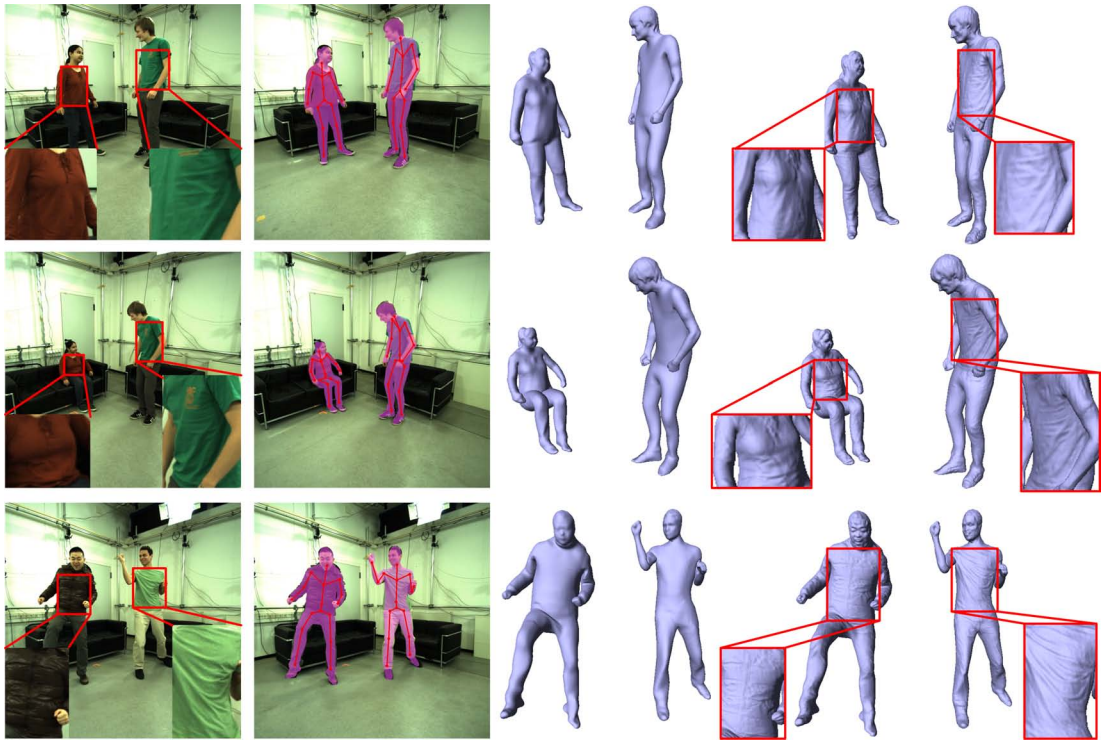


Figure 8.5: Performance capture results of our algorithm on real world sequences. Left to right: one of the two input images, segmentation and tracked skeleton as an overlay, 3D geometry after skeletal pose estimation, 3D geometry after surface refinement.

(see Fig. 8.5), which would make tracking these sequences with previous approaches challenging. We also evaluate our method on a synthetic data set. The pose for the first frame is initialized manually, followed by the local optimization described in Sec. 8.5.2. The mask image for the first frame is generated using a segmentation tool Rother *et al.* (2004).

The first sequence (Fig. 8.5, top) contains two people who initially are standing and talking, and then start to dance. Our algorithm successfully evaluates the pose and reconstructs small details such as the folds in the shirts accurately from the stereo images. The second sequence (Fig. 8.5, middle) shows two actors in the process of sitting down on a couch, and is recorded with a moving camera. Even though the actors are in contact with the couch in the background and some partial occlusions take place, the motion and surface detail is reconstructed accurately. As the camera is moving, we only reconstruct the relative pose of the actors with respect to the camera (i.e., we do not distinguish between camera mo-

Figure 8.6: One of our performance capture results. Left to right: input video, reconstructed geometry, edited video with virtual logo added.

tion and actor motion). The third sequence (Fig. 8.5, bottom) shows two actors jumping and kicking. Even though the motions are very fast, the pose estimation is successful. Further, even though the left actor wears a highly specular jacket, the surface detail is reconstructed accurately by our method. This again highlights the importance of using a non-Lambertian BRDF, since a method based on Lambertian shading would fail to estimate surface detail accurately.

Scene Enhancement We use the tracked motion and refined surface of the scenes to modify the original footage from the stereo camera. As our geometry is spatio-temporally coherent, it is easy to add new textures on top of the original footage, or perform other modifications. Fig. 8.6 shows an example on one of the captured sequences (Fig. 8.5, top).

Quantitative Evaluation To evaluate our method quantitatively, we generated a synthetic data set consisting of 100 frames by rendering a captured sequence with a manually painted Phong-based material and texture onto a virtual stereo rig with a baseline of $\approx$4 cm under the environment lighting of St. Peter's Basilica Debevec (1998). Given the images, the initial model, and its BRDF, as well as the incident lighting, we ran our complete pipeline, including the scene flow estimation, foreground segmentation, motion tracking, and surface refinement. We then compared the results against the ground truth to quantify the accuracy of the skeletal motion and surface reconstruction (see Fig. 8.7). The evaluation shows that our algorithm is able to create a very accurate reconstruction of the synthetic scene, with an average joint position error of only 11.6 ± 5.09 mm, and average surface position and normal error of 6.92 ± 4.23 mm and 9.34 ± 7.7 degrees respectively.

To make sure that all parts of our pipeline are actually important, we also evaluated the approach on a real sequence of 500 frames by leaving out one or

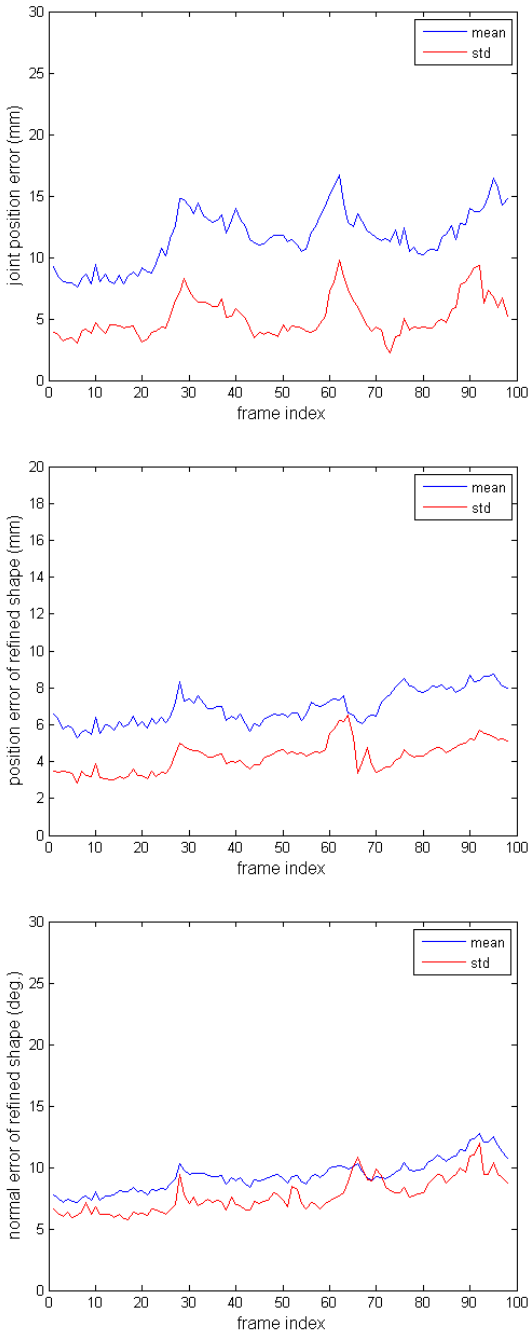


Figure 8.7: Quantitative evaluation on a synthetic sequence, showing mean and standard deviation for each frame: (a) joint position error, (b) vertex position error for refined shape, (c) normal direction error for refined shape.

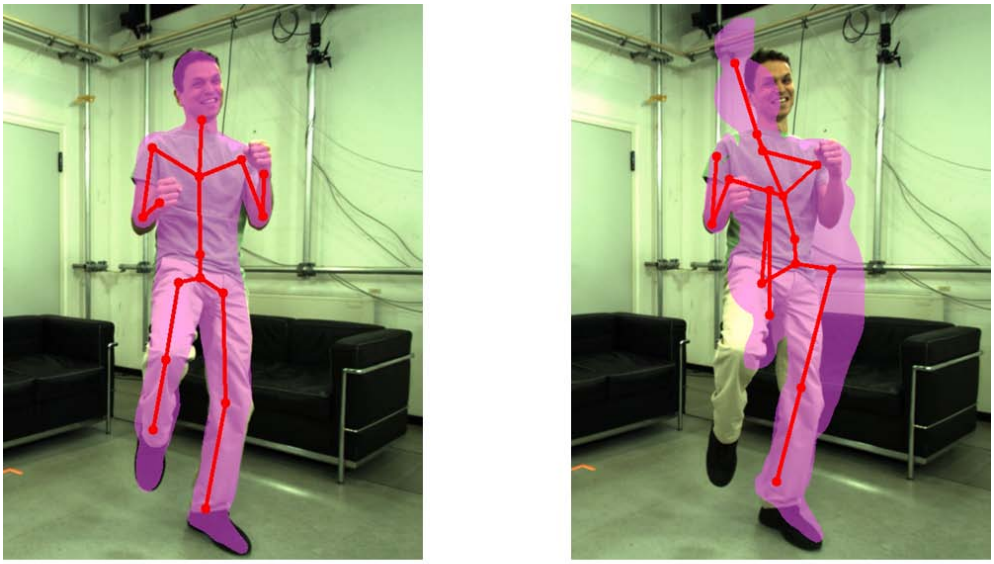


Figure 8.8: Comparison of our method with Gall *et al.* (2009): (a) our tracked skeleton, (b) tracked skeleton of Gall *et al.* (2009).

several stages of our pose estimation pipeline. Possible algorithmic components for the pose estimation pipeline are: (a) image segmentation, (b) scene flow constraints, (c) depth map constraints, (d) shading constraints, and (e) silhouette constraints. Using only (c), (c+d), or (a+b), the pose estimation fails to track the sequence completely. Using (a+c), or (a+c+d), the pose estimation is able to track the whole sequence, but some body parts get lost during tracking. Our pipeline, consisting of (a+c+d+e), is able to track the whole sequence correctly and performs best of all the combinations.

Comparison with previous methods We compared our tracking approach with the method described in Gall *et al.* (2009) for the real-world sequence shown in the bottom row of Fig. 8.5. As can be clearly seen in Fig. 8.8, the tracking method of Gall *et al.* (2009), which employs the silhouette and feature constraints, fails on this binocular data, while our method successfully estimates the correct pose.

We also compared our present method with our surface-based tracking method for binocular facial performance capture proposed in Chapter 7. Fig. 8.9 shows the results of tracking the template mesh over ≈ 200 frames for the real-world

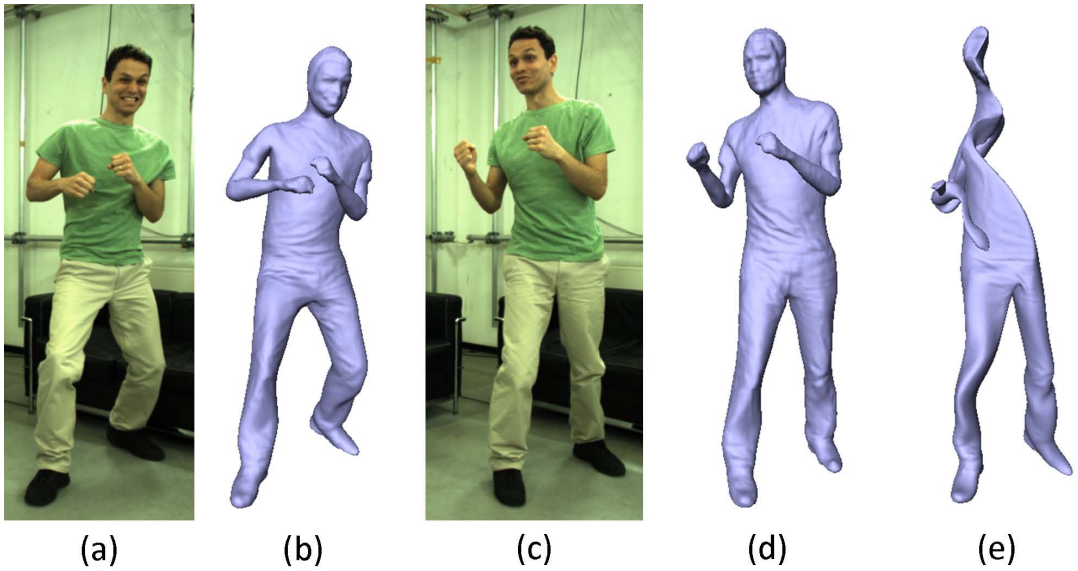


Figure 8.9: Comparison of our present method with surface tracking method proposed in Chapter 7: (a) first frame, (b) corresponding geometry, (c) 100th frame, (d) our reconstructed geometry using the present method, (e) reconstructed geometry using the method in Chapter 7.

sequence in the bottom row of Fig. 8.5. The method from Chapter 7, which only propagates mesh vertices by means of scene flow, clearly suffers from self-occlusions, motion estimation errors near boundaries, and the inability of the applied Laplacian regularization to deal with rotating motion. Our present method, on the other hand, builds on a model-based skeletal tracking that is much more robust to the articulated motion that is typical for full body tracking.

Run Time We ran our algorithm on a commodity PC with a dual-core 3GHz processor and 8GB RAM with a single threaded, non-optimized implementation. Scene flow calculation takes $\approx$3 min per frame. Motion tracking including foreground segmentation takes $\approx$2 min per frame. The final shape refinement step takes $\approx$1 min for a template mesh resolution of $\approx$80000 vertices. As these three steps are independent of each other, they can be pipelined into multiple threads or machines.

Discussion Our method successfully handles many challenging cases, including moving cameras, specular apparel, and partial occlusions. However, there are limitations to its use. As we use only a small-baseline stereo rig, some body parts

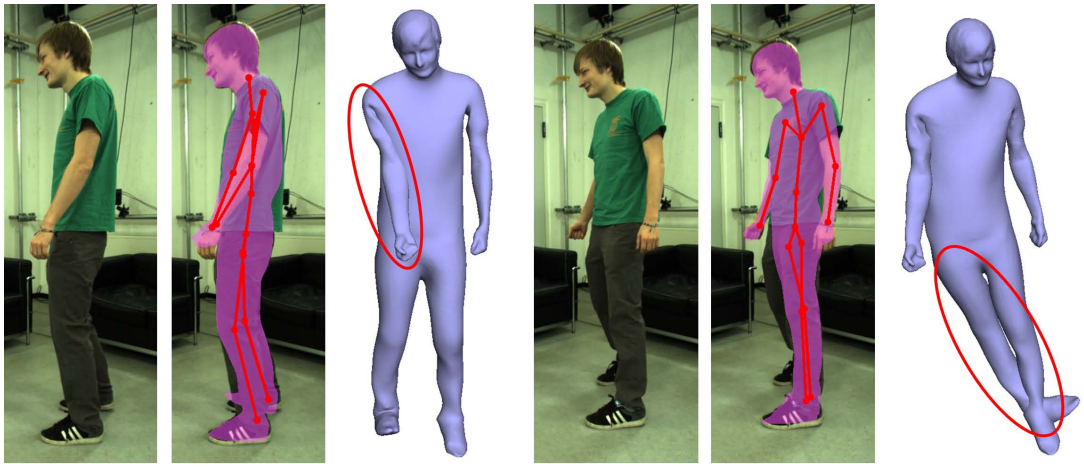


Figure 8.10: A failure case of our tracking method: (a) input frame in which an actor is turning away from the camera, (b) corresponding tracked skeleton, (c) corresponding tracked mesh, (d) input frame in which the same actor is turning back towards the camera, (e) corresponding tracked skeleton, (f) corresponding tracked mesh.

may be completely occluded in some frames. Our current local optimization scheme may fail to recover from these occlusions when the body parts appear again. Fig. 8.10 shows such a failure case for our method on a sequence with self-occlusions. Fig. 8.10 (a), (b) and (c) show one of the input images, the corresponding tracked skeleton from the camera view point, and the tracked mesh geometry for a frame in which an actor is turning away from the camera, thus occluding his right side. The tracked mesh makes it clear that the occluded arm is not tracked correctly and intersects the torso. Fig. 8.10 (d), (e) and (f) show results from the same camera view point at a later point in time where the right leg starts to reappear again. Both the tracked skeleton and mesh show an incorrect pose for the leg that was occluded in the previous frames. For the same reason, multiple interacting actors currently cannot be handled by our method. Occlusions could be handled by first detecting them and then using a global optimization for the occluded parts to make sure they are recovered correctly. The fact that occluded body parts do not have a correct pose during occlusion is not a major concern since our primary interest lies in the geometry visible from the perspective of the stereo camera. Nevertheless, recovering a reliable pose for occluded parts is an important open problem and may be relevant for some applications.

Extending the current method to outdoor performance capture is another interesting direction for future work. While our shape refinement algorithm is able to generate detailed geometry for most surfaces, it may fail for saturated and over-exposed highlights where no information can be extracted. Topological changes cannot be handled either, as we assume a constant connectivity and topology. Even though the output of our algorithm is spatio-temporally coherent (i.e., it has a constant connectivity and mesh topology), the shape refinement currently does not account for minor motion of garments such as a shifting shirt, which may lead to slight swimming artifacts in the range of 1-2 cm when rendering virtual textures in the original video. This could be improved by performing an additional scene flow-based alignment between the virtual actor and the current input images and performing an additional adaptation of the actor to the foreground segmentation to capture cloth motion.

8.8 Conclusion

We have presented a novel performance capture algorithm that reconstructs detailed human skeletal motion and space-time coherent surface geometry from a potentially moving, low-baseline stereo camera rig. It is able to track skeletal motion and detailed surface geometry of one or more actors in uncontrolled environments by exploiting BRDF information, scene illumination, and background segmentation. With our approach, we are able to produce high-quality results from a simple stereo camera setup; they approach the quality of results previously only achievable with complex setups containing 10 or more cameras. We believe that our method is a step towards making full-body performance capture available for wider use, such as on-set performance capture without additional hardware, video editing, and the creation of virtual actors.

Part IV

Other Applications

The ability to estimate the lighting, the reflectance, and the geometry from image or video input enables many applications. In this part, we list two methods that build on the techniques proposed in the previous parts of the thesis. One is relightable performance capture, which estimates not only detailed spatio-temporally coherent dynamic scene geometry, but also a spatially-varying surface reflectance model. The estimated dynamic geometry and surface reflectance enable the captured performance to be relit under a new environment. The other application introduced in this part is monocular facial performance capture. This method builds on the technique proposed in Chapter 7, but takes a step further by using only a monocular video input.

Chapter 9

Relightable Performance Capture and Monocular Facial Performance Capture

9.1 Relightable Performance Capture

9.1.1 Introduction

Capturing real performances of human actors and reproducing them in virtual environments has been one of the grand challenges in computer graphics and computer vision in the last few decades. Recent advances in marker-less multi-view video based capture methods have made it possible to reconstruct the motion, geometry and texture of actors de Aguiar *et al.* (2008); Gall *et al.* (2009); Vlasic *et al.* (2008), and create new motions from the performances Stoll *et al.* (2010) from arbitrary viewpoints. In previous chapters, we show how to achieve a high-quality dynamic scene reconstruction under a general environment. However, reconstructing a realistic appearance of the models is still challenging.

So far, most methods for rendering captured performances resort to projective texturing from the input video frames, e.g., Matusik *et al.* (2000); Starck & Hilton (2007). However, rendering a captured scene under new illumination has not yet been feasible. To overcome this limitation, some dynamic scene reconstruction methods estimate a spatially-varying BRDF of the scene model by filming such scenes under calibrated studio lighting Theobalt *et al.* (2007). Other approaches combine scene reconstruction with image-based relighting techniques Einarsson *et al.* (2006); Matusik *et al.* (2002) by recording under an advanced controllable light stage. However, these approaches are fundamentally limited by the fact that

Figure 9.1: Several images of a reconstructed real-world performance rendered from novel viewpoints and under a novel lighting condition. (Environment map courtesy of Paul Debevec.)

they require complex, expensive, and controlled camera and light setups which only exist in controlled studio environments. For many practical animation productions in movies or games, or for recording of 3D video, the requirement for controlled calibrated lighting is a major hindrance. Essentially, movie professionals would like to capture performances that can be relit directly on an arbitrary movie set where lighting can be arbitrary and vary greatly over time. The importance of this becomes clearer if one considers that many production sets are actually outdoors.

In this section, based on the techniques proposed in previous chapters, we introduce a new performance capture method that reconstructs not only detailed spatio-temporally coherent dynamic scene geometry, but also a spatially-varying parametric surface reflectance model of a human from a sparse set of multi-view video recordings under general uncontrolled illumination. Estimating a relightable performance under general lighting is a chicken-and-egg problem, as neither shape, illumination, nor surface reflectance are known in the beginning. Taking a strategy similar to previous chapters, we resort to a coarse-to-fine reconstruction scheme that eventually outputs highly detailed dynamic scene geometry, an all-frequency model of incident illumination, and a parametric spatially-varying BRDF model for the moving surface. In this way, we are able to keep the individual sub-estimation problems feasible in terms of computation time and the signal processing theory of inverse rendering. Plausibly relit performances can be created from multi-view video footage under general unknown lighting. Fig. 9.1 shows a reconstructed real-world performance rendered from novel viewpoints and a novel lighting condition. The work presented here was published in Li *et al.* (2013). The author of this thesis contributed to this work through

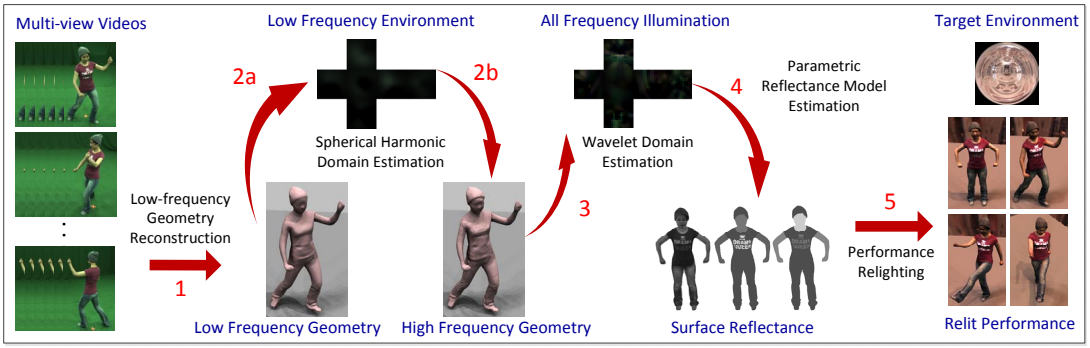


Figure 9.2: Overview of the method, illustrating the steps for geometry reconstruction (1 and 2b), lighting estimation (2a and 3), reflectance estimation (4), and final performance relighting (5).

his expertise on low-frequency lighting estimation and shading-based shape refinement, as described in Chapter 5. The new method for estimating the BRDF and the high-frequency incident lighting are not contributions of this thesis. To follow, we therefore briefly illustrate how the inverse rendering concepts of this thesis are used, and we refer the reader to Li *et al.* (2013) for details on the other algorithmic aspects.

9.1.2 Method

Input to our system is a multi-view video sequence of a moving actor captured using a sparse set of N_c synchronized cameras running at a standard frame rate (N_c typically between 8 and 9). The cameras are expected to be geometrically calibrated. They are also assumed to have linear response (if exact response curves are available, they are used), and color matching across views is performed during pre-processing. However, we do not impose strict requirements concerning the scene illumination. We need a rigged 3D shape model template comprising a skeleton, a triangle mesh surface model, and skinning weights for each vertex, which connect the mesh to the skeleton. The surface mesh of the actor can be generated using a laser scanner or image based 3D reconstruction techniques and is rigged using standard tools.

Given this input data, our algorithm reconstructs high resolution spatio-temporally coherent geometry, surface reflectance, and incident illumination. As shown in Fig. 9.2, we first reconstruct coarse geometry, reflectance and illumination and gradually refine the estimations over several steps of the pipeline. In

detail, our reconstruction approach consists of the following four main steps:

- **Low-frequency geometry reconstruction.** Similar to Chapter 5, we use a 3D model to track the articulated motion of the actor in the videos and reconstruct a dynamic geometry sequence using the marker-less motion capture approach of Gall *et al.* (2009) (Fig. 9.2 step 1). We purposefully use a smoothed geometry model for motion tracking, where the high-frequency surface detail has been removed. This yields an initial spatio-temporally coherent low-frequency 3D model of the dynamic scene.

- **Low-frequency illumination estimation and high-frequency geometry refinement.** Based on the technique proposed in Chapter 5, we use the initial geometry from the marker-less motion tracking approach to estimate the incident illumination in a spherical harmonic basis, as well as an initial piecewise constant diffuse surface albedo map (Fig.9.2 step 2a). From this, we refine the geometry of our model to recover the time-varying high-frequency geometry component (Fig. 9.2 step 2b).

- **All-frequency illumination estimation.** We use the detailed geometry to refine the previous low-frequency estimate of incident illumination into a more detailed representation, modeling high frequency lighting effects. We solve an inverse rendering problem to find a wavelet-based all-frequency representation, regularized by our initial low-frequency illumination model and the spatial total variation of the illumination (Fig. 9.2 step 3). The lighting is estimated in each color channel.

- **Reflectance reconstruction.** The estimated high-frequency incident illumination model is used to solve for a reflectance model that represents high-frequency reflectance effects. We estimate a spatially-varying parametric BRDF model with a temporally-varying diffuse component (Fig. 9.2 step 4). As we make no *a priori* assumption about the illumination, our algorithm estimates reflectance under general and uncontrolled illumination, even to a certain extent under time-varying illumination.

- **Performance relighting.** Given the reconstructed dynamic shape and reflectance information of the input performance, we now render the actor's performance from arbitrary viewpoints under arbitrary novel illumination conditions (Fig. 9.2 step 5).

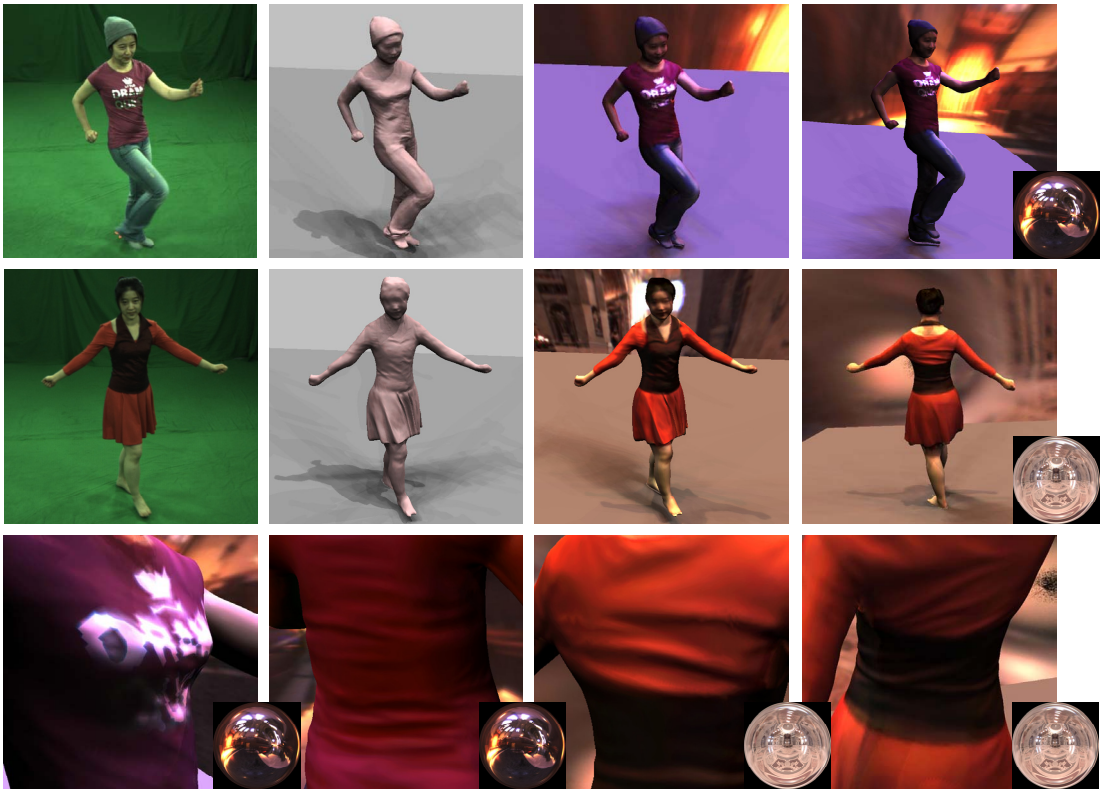


Figure 9.3: Results of our algorithm on different input scenes. The first two rows show, from left to right, single input camera view, reconstructed geometry, relit performance from input camera view, and relit performance from novel view. The bottom row shows closeups on the characters, highlighting the reconstructed and relit fine geometry details and specular reflections (Environment maps Grace Cathedral and St. Peter's Basilica courtesy of Paul Debevec).

9.1.3 Results

We reconstructed geometry, reflectance, and illumination from multi-view video sequence. The sequences *kungfu* (Fig. 9.1 and Fig. 9.3, row 1) and *dance* (Fig. 9.3, row 2) were reconstructed with 9 cameras in our multi-view video studio. We used general arbitrary studio lighting which was not controlled or designed in any specific way. Cameras recorded at a resolution of 1296×972 pixels, and at a frame rate of 45 fps; they were placed in a roughly circular arrangement around the scene.

The results show that this algorithm is able to capture geometry, reflectance and illumination in a believable way. Plausibly relit performances can be ren-

dered from arbitrary new viewpoints and under novel environment lightings. In the results, small-scale time-varying surface detail is plausibly captured and relit, such as folds in clothing (Fig. 9.3 bottom row). In particular, high-frequency reflectance effects are captured: for example, the specularity in the print of the t-shirt in the *kungfu* sequence, and the slight specularities in the skin and other fabrics in the *dance* sequence. This approach does not assume static per-vertex reflectance, and thus can also handle changing facial expressions or shifting apparel to a certain extent.

9.2 Dynamic face geometry from monocular video

9.2.1 Introduction

In Chapter 7, we presented an approach for detailed facial performance capture from binocular video. However, if only a monocular video is available, 3D face models of a quality level needed for movies and games cannot yet be captured. In this section, we briefly discuss a research project that aims to push the boundary and application range further and move towards monocular video. We introduce a new method to automatically capture detailed dynamic face geometry from monocular video filmed under general lighting. It fills an important algorithmic gap in the spectrum of facial performance capture techniques between expensive controlled setups and low-quality monocular approaches. It is a step towards democratizing face capture technology for everyday users with a single inexpensive video camera. The work briefly described here was published in Garrido *et al.* (2013). The author of this thesis contributed to this work with an adaptation of the dynamic shape refinement method from Chapter 7 to the single camera case. The main contribution of Garrido *et al.* (2013) on blend shape modeling and parameterization, monocular shape tracking, and flow-based surface refinement and stablization are not part of this thesis. To follow, we therefore only briefly review these parts of the project and focus on results illustrating the usefulness of shading-based shape refinement in this setting. More detail can be found in Garrido *et al.* (2013).

Our approach consists of several algorithmic components that are joined with state-of-the-art 2D and 3D vision and graphics techniques adapted to monocular video. In a one-time preparatory step, we create a personalized blend shape model for the captured actor by transferring the blend shapes of a generic model to a single static 3D face scan of the subject. Then, in the first step of our

Figure 9.4: Two results obtained with our method. Left: The input video. Middle: The tracked mesh shown as an overlay. Right: Applying texture to the mesh and overlaying it with the input video using the estimated lighting to give the impression of virtual face make-up.

automatic algorithm, we track a sparse set of 2D facial features throughout the video by adapting a probabilistic face tracking method that is regularized by a parametric 3D face model, learned once from a training set. The accuracy of the 2D landmark localization is improved by a feature correction scheme that uses optical flow for tracking correction. After 2D landmark tracking, we obtain the blend shape and pose parameters of the personalized 3D face model by solving a constrained quadratic programming problem at every frame. To further refine the alignment of the face model to the video, a non-rigid, temporally coherent geome-

Figure 9.5: Algorithm overview: Left to right: (a) Input video frame, (b) sparse feature tracking, (c) expression and pose estimation using a blend shape model, (d) dense expression and pose correction, (e) shape refinement.

try correction is performed using a multi-frame variational optical flow approach. Finally, a shape-from-shading-based shape refinement approach, inspired by the previous chapters, reconstructs fine scale geometric face detail after estimating the unknown incident lighting and face albedo.

This approach is one of the first in the literature to capture long sequences of expressive face motion for scenarios where none of these other methods are applicable. With the temporally coherent dynamic geometry obtained by our method, advanced video editing can be performed on the input video, for instance, by adding virtual face textures to the video. Fig. 9.4 shows two of the results from our method.

9.2.2 Method

Our method uses as input a single video of a face captured under unknown lighting. It is composed of four main computational steps:

- **Personalized face model creation.** We construct a customized parametric 3D blend shape model for every actor, which is used to reconstruct all sequences starring that actor.

- **Blend shape tracking.** In a first step, we track 2D image features throughout the monocular video by combining sparse facial feature tracking with automatic key frame selection and reliable optical flow; see Fig. 9.5 (b). From the established sparse feature set, we estimate a global 3D transformation (head pose) and a set of model parameters (facial expression) for the blend shape model; see Fig. 9.5 (c).

- **Dense tracking correction.** Next, we improve the facial expression and head pose obtained from sparse blend shape tracking by computing a temporally coherent and dense motion field in video and correcting the facial geometry to obtain a more accurate model-to-video alignment; see Fig. 9.5 (d).

- **Dynamic shape refinement.** In a final step, we reconstruct fine-scale, time-varying facial detail, such as wrinkles and folds. We do this by estimating the unknown lighting and exploiting shading for shape refinement; see Fig. 9.5 (e). This is based on an extension of the method from Chapter 7 to monocular video input where lighting is estimated over a span of subsequent video frames.

Since the dynamic shape refinement is the step that exploits inverse rendering for fine geometric detail estimation, we will elaborate on this step next.

9.2.3 Dynamic Shape Refinement With Monocular Video Input

In this step, we capture fine-scale, possibly time-varying surface detail, such as emerging or disappearing wrinkles and folds, which are not yet represented in the tracked mesh by the previous steps. The approach is based on the shape-from-shading framework under general unknown illumination that was proposed in Chapter 7. Based on an estimate of low-frequency geometry, the method first estimates the unknown incident lighting and the surface reflectance at the current time step and then uses the known lighting and reflectance to deform the geometry such that the rendered shading gradients and the image gradients agree. Essentially, the method inverts the rendering process to reconstruct the scene, which is easier in a setting with multiple cameras, where the fact that a surface is seen from several viewpoints constrains the solution space better.

To adjust this approach to the monocular case, we estimate the unknown illumination from a larger temporal baseline to compensate for the lack of additional cameras. In our setting, we assume that the illumination conditions do not change over time, yet a ground truth light probe may not be available. Therefore, we first estimate lighting, albedo and refined surface geometry of the tracked face mesh for the first 10 frames of every video using the exact same approach as in Chapter 7. Since in the monocular case the estimation is much more under-constrained and error-prone, we only use this result as an initialization. In a second step, we jointly use the initial albedo and fine scale geometry to estimate

a single environment map that globally fits all time steps. We then use this static light environment and estimate the dynamic geometry detail at each time step. The result of dynamic shape refinement is the final refined face mesh.

9.2.4 Results

We evaluate the performance of our approach on multiple video sequences of different actors with lengths ranging from 560 (22s) to 1000 frames (40s). Three videos are recorded with a Canon EOS 550D camera at 25 fps in HD quality.

Face capture results. The first two results are obtained by testing the algorithm on the face sequences from Chapter 7, which are recorded under uncontrolled indoor lighting. Here, we only use one camera input for our method and need one extra frame from the second camera for the blend shape model creation. Results for the first sequence, featuring very expressive gestures and normal speech, are shown in Fig. 9.6. The figure shows that we are able to faithfully capture very challenging facial expressions, even for gestures that are not spanned by the blend shape model, e.g. the right column. Fig. 9.7 shows a result for a second sequence of around 620 frames (25s), featuring fast and expressive motion and a high level of surface detail. Also for this sequence, our results capture the facial geometry and motion with high detail.

Fig. 9.8 shows an additional result for a third sequence, newly recorded under similar conditions as the first two. The sequence depicts a recitation of a theatrical play and is extremely challenging due to its length of 1000 frames (40s), its diversity of facial expressions, and fast and shaky head motion. The overlays in the figure show that we are able to estimate the X- and Y-component of the head pose very accurately and retrieve very subtle facial expressions, demonstrating the applicability of our method for demanding real-world applications.

Virtual face texture. Our capturing process introduces hardly any perceivable drift, so it is well suited for video augmentation tasks, such as adding virtual texture or tattoos[1]; see Figs. 9.4 and 9.6. To this end, we render the texture as a diffuse albedo map on the moving face and light it with the estimated incident illumination. The texture is rendered in a separate channel and overlaid with the input video using Adobe Premiere. Our detailed reconstruction and lighting of the deformation detail is important to make the shading of the texture correspond to the shading in the video, giving an impression of virtual make-up.

[1] www.deviantart.com/

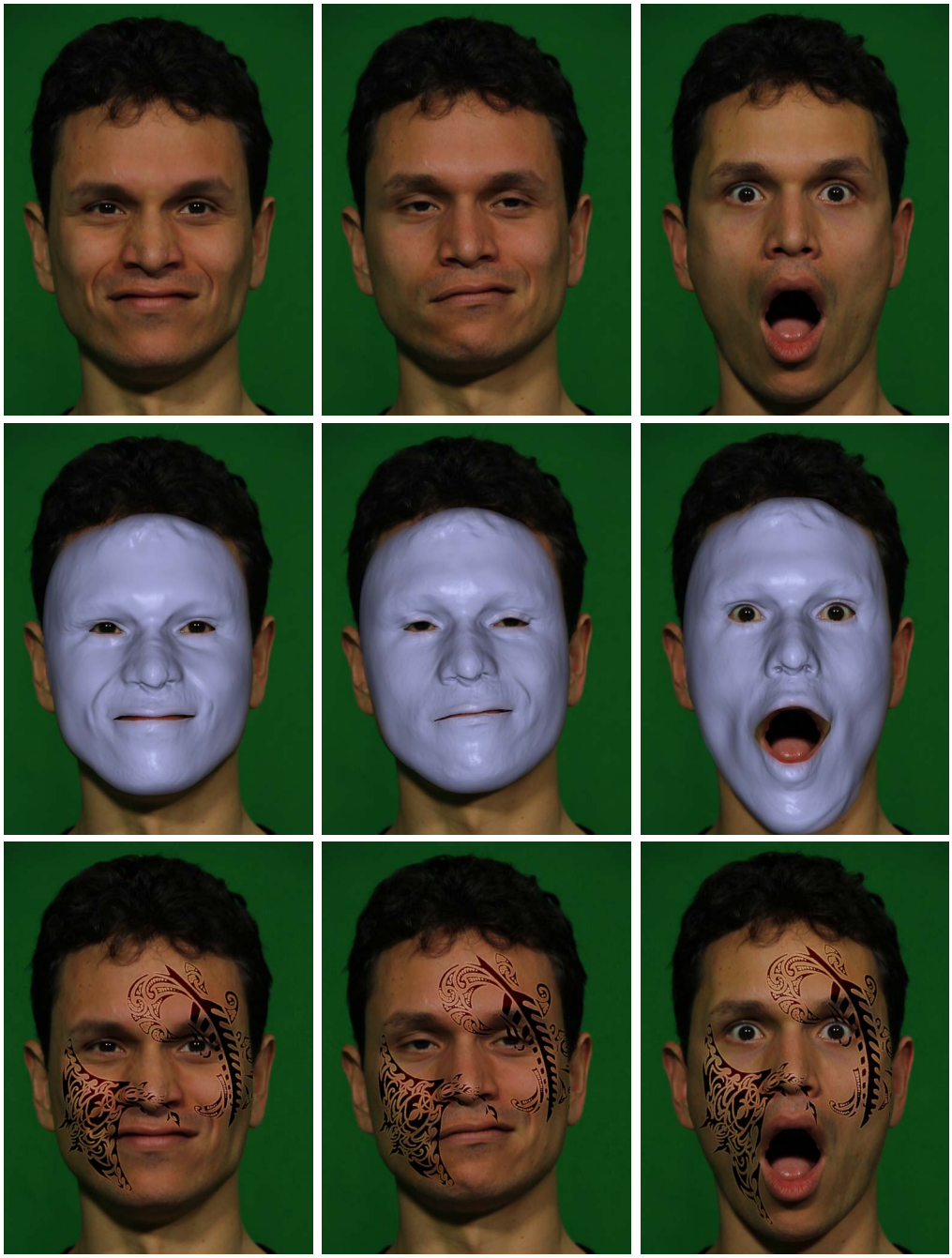


Figure 9.6: Monocular reconstruction results for expressive facial motion. Top to bottom: The input frame, the corresponding blended overlay of the reconstructed mesh, a 3D view of the mesh, an example of applying virtual face texture using the estimated geometry and lighting.

Figure 9.7: Monocular facial performance capture results for very expressive and fast facial gestures and challenging head motion for up to 1000 frames.

Figure 9.8: Monocular facial performance capture results for very expressive and fast facial gestures and challenging head motion for up to 1000 frames.

Chapter 10

Conclusions

In this thesis, we advance the theory and practice of static and dynamic scene reconstruction from images or videos. The existing technologies are constrained, requiring a lot of prerequisites to succeed. For static scene reconstruction, previous methods have difficulties reconstructing high-frequency detail in a general environment. For dynamic scene reconstruction, marker-less performance capture methods, which are able to reconstruct dynamic geometry from a handful of video recordings, heavily rely on specified setups, e.g. controlled lighting and controlled background, and they require a large number of cameras. These requirements restrict the technology to working only in a studio environment, and prevent it from being broadly used by the movie industry or by ordinary consumers. One of the major reasons for these limitations is that the information in the images or videos has not been fully exploited or appropriately modeled. Appearance changes have been treated as artifacts rather than signals. To obtain a stronger model, in this thesis, we look into the physical process of how the image or video is generated in a general environment, and investigate inverse rendering for reconstructing both static and dynamic scenes. In this way, we proposed techniques for scene reconstruction which overcame the limitations of existing methods and achieved high-quality scene reconstruction in a general environment by using as few cameras as possible.

We approached our goal step by step, overcoming existing limitations and relaxing constraining assumptions. Firstly, we demonstrated the concept of exploiting inverse rendering for 3D reconstruction in static scenes under general illumination, but using multi-view images and assuming Lambertian surface reflectance with uniform albedo. Specifically, in the first part of this thesis, we developed a method that combines multi-view stereo and shading cues to obtain

a high-quality 3D reconstruction for static objects under general unknown lighting. By assuming Lambertian reflectance with uniform albedo, we simplified the inverse rendering problem to two unknowns, namely the lighting and the geometry. Our method starts with a coarse 3D model reconstructed from multi-view stereo. Then, this coarse model is used to estimate the lighting. With the lighting estimated, the coarse geometry is refined by minimizing the shading error to achieve a final high-quality reconstruction. Our high-quality results prove the validity of the concept of using inverse rendering for scene reconstruction, so far for static scenes.

In part II, we took a step in the temporal domain and investigated inverse rendering for capturing full-body performance with a multi-view camera setup. The goal in this part was to relax the constraints of traditional performance capture methods, which required controlled lighting and background. To achieve this goal, two steps were taken to reconstruct detailed models of dynamic scenes with spatially varying Lambertian reflectance from multi-view video footage in a general environment. For the first step described in Chapter 5, we developed a dynamic shape refinement method, which is based on shading cues in the video and also used a temporal geometry prior relative to previous time steps, to capture millimeter-scale surface structure under general and unknown illumination. Here, we still assume Lambertian reflectance but allow the surface albedo to be spatially varying. With an off-the-shelf performance capture method of Gall *et al.* (2009) for low-frequency geometry estimation, our method is able to capture high-quality dynamic shape with true fine geometric detail under general and unknown illumination, which was previously only possible with specially engineered lighting. However, this method is still constrained by the employed low-frequency geometry estimation method, which requires constant lighting and a green-screen background. In Chapter 6, we therefore proposed a new performance capture method which works under general, unknown and changing illumination, and in front of a general scene background instead of a green screen. We achieved this by exploiting inverse rendering for model-based skeletal motion estimation, as well as for combined low frequency and high frequency 3D geometry reconstruction. Skeletal pose and low-frequency geometry are estimated with a joint optimization for the unknown lighting. Then, the low-frequency geometry is further optimized by our dynamic shape refinement method to capture fine-scale, non-rigid surface deformation. This whole method works in a general environment, but still requires eight or more cameras and assumes a discrete set of Lambertian reflectances on the surface.

In part III, we made efforts to reduce the number of cameras required. We presented two performance capture methods which use as few as two cameras for the capture and achieve high-quality dynamic scene reconstruction in a general environment. In Chapter 7, we proposed a method to reconstruct facial performance whose reconstruction quality comes close to much more complex multi-camera approaches. This method first captures a low-frequency face model for each frame by deforming a face template with a scene flow constraint and following a motion correction step. Then, improved inverse rendering techniques are performed to estimate the lighting, the reflectance and the geometry. This method is the first purely passive technique that enables high-quality and spatio-temporally coherent facial performance capture from only two cameras, while being applicable in uncontrolled or even changing lighting scenarios, e.g. outdoors. In Chapter 8, a new full-body performance capture method, which only takes input from a binocular stereo rig, was presented. This method succeeds in capturing the full body skeletal motion and detailed surface geometry of one or more actors wearing general apparel in a general environment. The Lambertian assumption in previous chapters is extended here to a general BRDF. This is one of the first performance capture methods to exploit the full BRDF information and scene illumination for accurate pose tracking and surface refinement in general scenes.

Finally in part IV, two applications were introduced which were made possible by the techniques proposed in this thesis. One is relightable performance capture under general illumination, which takes the captured dynamic geometry from the method proposed in Chapter 5 as input, and estimates the all-frequency lighting and full BRDF parameters, including diffuse and specular reflectance of the surface. The estimated reflectance, as well as the high-quality dynamic geometry, allows the captured dynamic scenes to be relit under a novel illumination condition. In the second application, we demonstrate monocular facial performance capture, which takes the binocular method proposed in Chapter 7 a step further. In this application, a coarse 3D model is first obtained by blend-shape-based face tracking, followed by an optical-flow-based geometry correction. Then, a shape refinement method similar to that in Chapter 7 is employed to add the high-frequency detail.

In summary, this thesis takes several important steps towards the goal of static and dynamic scene reconstruction in a general environment. Contributions are made in terms of reducing the constraints on lighting, background and required number of cameras. The key contribution lies in the investigation of light transport in the scene for improved reconstruction of lighting, reflectance

and geometry. By looking into the rendering process, additional insights for reconstruction are gained; the shading cues, which act as rich information in the images, are utilized; the environment lighting in the scene, which is an important signal, is theoretically modeled. We believe that, with the work proposed in this thesis, a new path will be opened up for applying marker-less performance capture, not only in industrial movie and game productions, but also in the daily-life scenarios of average consumers.

10.1 Future Directions

Despite the algorithmic improvements described in this thesis, scene reconstruction in a general and arbitrary environment is still far from being solved. There are still many challenges remaining. In the following, we list a few of them and discuss future work.

10.1.1 Improved Modeling and Inversion of Light Transport

In this thesis, we employed the reflection equation parameterized by spherical harmonics for inverse rendering and for scene reconstruction. The spherical harmonic basis functions can represent the low-frequency signal of lighting and reflectance well, for instance, when representing the clamped cosine function. However, if the signal has quite a lot of high-frequency components, e.g. in the case of representing the specular reflectance function, higher orders of spherical harmonics are needed. But increasing the order of spherical harmonics comes at the price of higher computational complexity to solve the inverse rendering problems. Besides, a too high order in the spherical harmonics parameterization will cause ringing artifacts in the spatial domain. As the spherical harmonics are just one type of basis functions, other basis functions like wavelets, which have advantages for representing high-frequency signals, can be investigated. In fact, in the field of rendering, the benefit of using wavelets to represent the reflection equation has been demonstrated Ng *et al.* (2004). We also took a further step to estimate the full-frequency lighting by using wavelets (see Chapter 9). However, due to the additional complexity resulting from using a wavelet parameterization, research is needed to make use of it for geometry reconstruction. In addition to a wavelet representation, other basis functions can also be employed, if they can lead to a more compact representation.

The reflection equation based on the environment mapping, which was used in this thesis, is an approximation of the full rendering equation that describes the global process of light transport. There are several simplifying assumptions underlying this approximation. One comes from the environment mapping for the lighting, which assumes infinitely distant lights. This works well if the size of the object is relatively small in comparison to the distance between the lights and the object. But for indoor scenes, e.g. a full-body performance capture in a room where the space is confined, this assumption may not necessarily hold when the actor approaches the light source. Reconstructing a location-dependent lighting mapping may help to solve this problem. Another assumption is that light transport is characterized by the first bounce, so interreflections are ignored during inverse rendering. To solve this problem, more accurate rendering models, i.e. the full and non-simplified rendering equation, can be used for reconstruction. However, this could lead to a vastly increased complexity and a much bigger set of unknowns. For instance, in order to solve inverse rendering problems, the synthesized images simulated by the rendering equation are needed. However, simulating a physically correct rendering result, e.g. using Monte Carlo algorithms Lafortune (1996), will need a lot more time, not to mention that the optimization in inverse rendering requires thousands of iterations for such simulation. Besides, the physically correct lighting cannot be simply represented by one environment map but should be the actual 3D surroundings, which will result in a huge space of unknowns to solve.

With the light transport model chosen, inverse problems need to be solved to estimate the lighting, the reflectance or the geometry from the captured images or videos. Due to the insufficient samples and large space of unknowns, the inverse rendering problems are usually ill-posed and non-trivial to solve. In this thesis, we firstly reconstruct coarse 3D geometry by multi-view stereo in Chapter 4, by skeletal motion estimation in Chapters 5, 6 and 8, or by surface mesh tracking in Chapter 7. Then, with this initialization for the surface geometry, we perform joint optimization over the lighting, the reflectance and the geometry. Currently, the color ambiguity between lighting and reflectance has not been solved totally. In this thesis, either we assume a neutral lighting, the color of which is white, or we impose a constraint on the reflectance color as input. Solving this color ambiguity generally needs additional constraints or priors to be imposed on the lighting or the reflectance. Moreover, we are currently not able to handle arbitrary, spatial-varying surface reflectance, e.g. a surface where each surface point has a unique reflectance value. The reason for this is that it will result in a huge space of

unknowns, which are difficult to optimize. So in this thesis, we either assumed a piecewise uniform albedo map (Chapter 5), or that the reflectance of the surface could be represented by a set of reflectance clusters (Chapter 7). Although these assumptions work well for the specific objects in question, a better prior on the reflectance is needed for more complex scenarios. For the environment lighting, additional priors can also be assumed. For example, the lighting itself can be approximated by a captured light probe image, and this light probe image can in turn be constrained by image priors that facilitate reconstruction, as studied by Huang & Mumford (1999). We believe an effective prior on the lighting will improve its estimation. In this thesis, we also assume an initial coarse solution for the 3D geometry can be obtained. However, for some scenarios, e.g. with a monocular video input of a general scene, the initial coarse geometry may not be easy to obtain. In this case, better priors on lighting and reflectance, which have been demonstrated to be helpful in the shape from shading framework Barron & Malik (2012), are needed. We believe with better priors on lighting, reflectance and geometry, general scenes can be captured even with a monocular video input.

10.1.2 Reconstructing Complex Dynamic Scenes

When reconstructing complex dynamic scenes, there are some relevant limitations to the current methods proposed in this thesis. One limitation is related to occlusions, which can cause a problem for several methods in this thesis. The method from Chapter 6, which relies on a multi-view setup, can handle slight occlusion, for instance, the case where one of the cameras is occluded by the dynamic background. But generally it will become more difficult when more cameras are occluded. Moreover, the method is applicable only towards the reconstruction of one actor. It will become challenging to capture more actors in the scene, as the occlusions between actors become severe. Inspired by Liu *et al.* (2011), one potential solution is to first segment each image into multiple regions assigned to different actors, and then to reconstruct the performance of each actor by using only the shading information from the corresponding regions. However, the more occlusions there are, the more the information in the images that is useful for performance capture will be reduced. This may make the reconstruction problem far more underconstrained, especially as more actors are present in the scene. For the method proposed in Chapter 8, occlusion handling will be even more challenging, since there are only two camera views looking at the scene from a low baseline setup. As also discussed in Chapter 8, under this setting,

self-occlusions can cause some body parts to be completely occluded. In these cases of self-occlusions, no image-based constraints will be available to estimate the motion for those body parts. It will also cause problems if the occluded parts are later disoccluded, since enforcing continuity is difficult. To resolve this, future work may infer shape and motion of completely occluded parts from priors. To track the re-appearing parts, a body part detection step may be helpful. Similarly, it will also be challenging to handle interacting actors, or an actor interacting with other objects. For the facial performance capture method of Chapter 7, occlusions may not be as problematic as for the full body case, as long as the face always looks more or less straight into the camera. However, it will also have problems if the face is occluded by other objects, or if the face disappears and re-appears in the frame. Although off-the-shelf techniques exist for detecting a face and can be easily applied here, detecting and recognizing a face given the presence of other faces is still a non-trivial problem.

Another limitation of our methods is related to the fact that we assume a constant mesh connectivity and topology for the template mesh of an actor or a face. Thus, changes of apparent topology during capture, which could be, for instance, due to putting on or taking off clothes, or putting on a face mask, cannot be handled by the methods of Chapters 6, 7 and 8. A potential solution could be to use several layers of different templates for different parts. For example, we can use a template layer for the inner body, and other template layers for coats, pants, etc. Template layers would also need to be reconstructed.

References

Alexander, O., Rogers, M., Lambeth, W., Chiang, M. & Debevec, P. (2009). The digital emily project: photoreal facial modeling and animation. In *ACM SIGGRAPH 2009 Courses*, 12:1–12:15, ACM. 32

Alldrin, N., Zickler, T. & Kriegman, D. (2008). Photometric stereo with non-parametric and spatially-varying reflectance. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008.*, 1–8. 26

Anuar, N. & Guskov, I. (2004). Extracting animated meshes with adaptive motion estimation. In *International Workshop on Vision, Modeling and Visualization*, 63–71. 31

Balan, A., Sigal, L., Black, M., Davis, J. & Haussecker, H. (2007). Detailed human shape and pose from images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. 28, 78

Baran, I. & Popović, J. (2007). Automatic rigging and animation of 3d characters. *ACM Transactions on Graphics*, **26**. 82, 83

Barron, J.T. & Malik, J. (2012). Color constancy, intrinsic images, and shape estimation. *European Conference on Computer Vision*. 166

Basri, R. & Jacobs, D.W. (2003). Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 218–233. 11, 44

Basri, R., Jacobs, D. & Kemelmacher, I. (2006). Photometric stereo with general unknown lighting. *Int'l Journal of Computer Vision*, **72**, 239–257. 26, 34

REFERENCES

Beeler, T., Bickel, B., Beardsley, P., Sumner, B. & Gross, M. (2010). High-quality single-shot capture of facial geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, **29**, 40:1–40:9. 27

Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R.W. & Gross, M. (2011). High-quality passive facial performance capture using anchor frames. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, **30**, 75:1–75:10. 32, 118

Belhumeur, P., Kriegman, D. & Yuille, A. (1997). The bas-relief ambiguity. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1060–1066. 26

Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H. & Gross, M. (2007). Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics*, **26**, 33:1–33:10. 31, 99

Blake, A., Zimmerman, A. & Knowles, G. (1986). Surface descriptions from stereo and shading. *Image Vision Comput.*, **3**, 183–191. 27

Blanz, V., Basso, C., Vetter, T. & Poggio, T. (2003). Reanimating faces in images and video. *Computer Graphics Forum (Proc. EUROGRAPHICS)*, **22**, 641–650. 31

Bo, L. & Sminchisescu, C. (2010). Twin gaussian processes for structured prediction. *Int'l Journal of Computer Vision*, **87**, 28–52. 28

Boivin, S. & Gagalowicz, A. (2001). Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 107–116, ACM, New York, NY, USA. 33

Borshukov, G., Piponi, D., Larsen, O., Lewis, J.P. & Tempelaar-Lietz, C. (2003). Universal capture: image-based facial animation for "the matrix reloaded". In *ACM SIGGRAPH 2003 Sketches*, 16:1–16:1, ACM. 32

Boykov, Y. & Funka-Lea, G. (2006). Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, **70**, 109–131. 89, 131

Bradley, D., Boubekeur, T. & Heidrich, W. (2008). Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. 26

Bradley, D., Heidrich, W., Popa, T. & Sheffer, A. (2010). High resolution passive facial performance capture. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, **29**. 1, 32, 59, 99, 105

Bray, M., Kohli, P. & Torr, P.H.S. (2006). Posecut: simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, 642–655. 29

Bregler, C., Malik, J. & Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *Int'l Journal of Computer Vision*, **56**, 179–194. 22, 28, 78, 82, 85

Brostow, G., Hernandez, C., Vogiatzis, G., Stenger, B. & Cipolla, R. (2011). Video normals from colored lights. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 2104–2114. 30

Brox, T., Bruhn, A., Papenberg, N. & Weicker, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, 25–36. 16, 85

Brox, T., Rosenhahn, B., Cremers, D. & Seidel, H.P. (2006). High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints. In *European Conference on Computer Vision*, 98–111. 29, 129

Brox, T., Rosenhahn, B., Gall, J. & Cremers, D. (2010). Combined region and motion-based 3D tracking of rigid and articulated objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 402–415. 29, 129

Cagniart, C., Boyer, E. & Ilic, S. (2010a). Free-form mesh tracking: a patch-based approach. In *IEEE Conference on Computer Vision and Pattern Recognition*. 29, 78

Cagniart, C., Boyer, E. & Ilic, S. (2010b). Probabilistic deformable surface tracking from multiple videos. In *European Conference on Computer Vision*. 59

REFERENCES

Carceroni, R.L. & Kutulakos, K.N. (2002). Multi-view scene capture by surfel sampling: from video streams to non-rigid 3D notion, shape and reflectance. *Int'l Journal of Computer Vision*, **49**, 175–214. 34

Cohen, M.F., Wallace, J. & Hanrahan, P. (1993). *Radiosity and realistic image synthesis*. Academic Press Professional, Inc., San Diego, CA, USA. 10

Cryer, J.E., Tsai, P.S. & Shah, M. (1995). Integration of shape from shading and stereo. *Pattern Recognition*, **28**, 1033–1043. 27

Davis, J.E., Yang, R. & Wang, L. (2005). Brdf invariant stereo using light transport constancy. In *IEEE International Conference on Computer Vision*, 436–443. 134

de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P. & Thrun, S. (2008). Performance capture from sparse multi-view video. *ACM Transactions on Graphics*, **27**, 1–10. 1, 3, 28, 29, 59, 72, 78, 147

Debevec, P. (1998). Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 189–198, ACM, New York, NY, USA. 137

DeCarlo, D. & Metaxas, D. (1996). The integration of optical flow and deformable models with applications to human face ahape and motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 231–238. 31

Deutscher, J., Blake, A. & Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1144–1149. 28, 78

Einarsson, P., Chabert, C.F., Jones, A., Ma, W.C., Lamond, B., im Hawkins, Bolas, M., Sylwan, S. & Debevec, P. (2006). Relighting human locomotion with flowed reflectance fields. In *Eurographics Symposium on Rendering*, 183–194. 147

Felzenszwalb, P.F. & Huttenlocher, D.P. (2004). Efficient graph-based image segmentation. *Int'l Journal of Computer Vision*, **59**. 64, 108

Fua, P. & Leclerc, Y.G. (1995). Object-centered surface reconstruction: Combining multi-image stereo and shading. *Int'l Journal of Computer Vision*, **16**, 35–56. 27

Furukawa, Y. & Ponce, J. (2009). Dense 3D motion capture for human faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1674–1681. 31, 99

Furukawa, Y. & Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 1362–1376. 26, 42, 43, 49

Fyffe, G., Hawkins, T., Watts, C., Ma, W.C. & Debevec, P. (2011). Comprehensive Facial Performance Capture. *Computer Graphics Forum*, **30**, 425–434. 32

Gall, J., Rosenhahn, B. & Seidel, H.P. (2008). *Human Motion: Understanding, Modelling, Capture and Animation*, chap. An Introduction to Interacting Simulated Annealing, 319–343. Springer. 28

Gall, J., Stoll, C., Aguiar, E., Theobalt, C., Rosenhahn, B. & Seidel, H.P. (2009). Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1746–1753. 1, 4, 28, 29, 60, 61, 66, 69, 74, 77, 78, 80, 81, 91, 132, 134, 139, 147, 150, 162

Ganapathi, V., Plagemann, C., Koller, D. & Thrun, S. (2010). Real time motion capture using a single time-of-flight camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, 755–762. 29

Garrido, P., Valgaert, L., Wu, C. & Theobalt, C. (2013). Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, **32**, 158:1–158:10. 5, 8, 152

Georghiades, A.S. (2003). Recovering 3-d shape and reflectance from a small number of photographs. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, 230–240, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland. 34

Goldman, D., Curless, B., Hertzmann, A. & Seitz, S. (2005). Shape and spatially-varying brdfs from photometric stereo. In *IEEE International Conference on Computer Vision*, vol. 1, 341–348. 34

Greene, N. (1986). Environment mapping and other applications of world projections. *Computer Graphics and Applications, IEEE*, **6**, 21–29. 33

Groemer, H. (1996). *Geometric Applications of Fourier Series and Spherical Harmonics*. Cambridge Univ. Press. 13

Guenter, B., Grimm, C., Wood, D., Malvar, H. & Pighin, F. (1998). Making faces. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 55–66, ACM, New York, NY, USA. 31

Haber, T., Fuchs, C., Bekaert, P., Seidel, H.P., Goesele, M. & Lensch, H.P.A. (2009). Relighting objects from image collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 627–634. 33, 34

Hartley, R. & Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press. 17, 102

Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J. & Seidel, H.P. (2009). Markerless motion capture with unsynchronized moving cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*. 29

Hernandez, C., Vogiatzis, G., Brostow, G.J., Stenger, B. & Cipolla, R. (2007). Non-rigid photometric stereo with colored lights. In *IEEE International Conference on Computer Vision*, 1–8. 60

Hernandez, C., Vogiatzis, G. & Cipolla, R. (2008). Muti-view photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 548–554. 27

Higo, T., Matsushita, Y., Joshi, N. & Ikeuchi, K. (2009). A hand-held photometric stereo camera for 3-d modeling. In *IEEE International Conference on Computer Vision*, 1234–1241. 33

Horn, B.K. (1970). *Efficient Rendering and Compression for Full-Parallax Computer-Generated Holographic Stereograms*. Ph.D. thesis, Massachusetts Inst. of Technology. 26

Horn, B.K. & Schunck, B.G. (1981). Determining optical flow. In *1981 Technical Symposium East*, 319–331, International Society for Optics and Photonics. 16

Huang, H., Chai, J., Tong, X. & Wu, H.T. (2011). Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. *ACM Transactions on Graphics*, **30**, 74:1–74:10. 31

Huang, J. & Mumford, D. (1999). Statistics of natural images and models. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, –547 Vol. 1. 166

Jin, H., Yezzi, A. & Soatto, S. (2000). Stereoscopic shading: integrating multi-frame shape cues in a variational framework. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 169–176. 27

Jin, H., Cremers, D., Yezzi, A. & Soatto, S. (2004a). Shedding light on stereoscopic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, **1**, 36–42. 27

Jin, H., Yezzi, A.J. & Soatto, S. (2004b). Region-based segmentation on evolving surfaces with application to 3d reconstruction of shape and piecewise constant radiance. In *European Conference on Computer Vision*, 114–125. 27

Jin, H., Cremers, D., Wang, D., Prados, E., Yezzi, A. & Soatto, S. (2008). 3-d reconstruction of shaded objects from multiple images under unknown illumination. *Int'l Journal of Computer Vision*, **76**, 245–256. 27

Kajiya, J.T. (1986). The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, 143–150, ACM, New York, NY, USA. 9, 40, 61

Kavan, L., Collins, S., Žára, J. & O'Sullivan, C. (2007). Skinning with dual quaternions. In *Symposium on Interactive 3D graphics and games*, 39–46. 24, 88

Kazhdan, M., Bolitho, M. & Hoppe, H. (2006). Poisson surface reconstruction. In *Symposium on Geometry Processing*, 61–70. 102

Kim, H., Wilburn, B. & Ben-Ezra, M. (2010). Photometric stereo for dynamic surface orientations. In *European Conference on Computer Vision*, 59–72. 30

Lafortune, E. (1996). Mathematical models and monte carlo algorithms for physically based rendering. *PhD thesis*. 165

Langer, M. & Bülthoff, H. (2000). Depth discrimination from shading under diffuse lighting. *Perception*, **29**, 649–660. 47

Leclerc, Y.G. & Bobick, A.F. (1991). The direct computation of height from shading. In *IEEE Conference on Computer Vision and Pattern Recognition*, 552–558. 27

Lee, C.S. & Elgammal, A. (2010). Coupled visual and kinematic manifold models for tracking. *Int'l Journal of Computer Vision*, **87**, 118–139. 28

Lensch, H.P.A., Kautz, J., Goesele, M., Heidrich, W. & Seidel, H.P. (2003). Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, **22**, 234–257. 33

Lewis, J.P., Cordner, M. & Fong, N. (2000). Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, 165–172, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. 82

Li, G., Wu, C., Stoll, C., Liu, Y., Varanasi, K., Dai, Q. & Theobalt, C. (2013). Capturing relightable human performances under general uncontrolled illumination. *Computer Graphics Forum (Proc. EUROGRAPHICS)*, **32**. 5, 8, 123, 126, 148, 149

Li, R., Tian, T.P., Sclaroff, S. & Yang, M.H. (2010). 3d human motion tracking with a coordinated mixture of factor analyzers. *Int'l Journal of Computer Vision*, **87**, 170–190. 28

Liao, M., Zhang, Q., Wang, H., Yang, R. & Gong, M. (2009). Modeling deformable objects from a single depth camera. In *IEEE International Conference on Computer Vision*, 167–174. 30

Liu, Y., Dai, Q. & Xu, W. (2010). A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE Transactions on Visualization and Computer Graphics*, **16**, 407–418. 26, 42, 49, 72, 74

Liu, Y., Stoll, C., Gall, J., Seidel, H.P. & Theobalt, C. (2011). Markerless motion capture of interacting characters using multi-view image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 29, 129, 131, 166

Matusik, W., Buehler, C., Raskar, R., Gortler, S.J. & McMillan, L. (2000). Image-based visual hulls. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, 369–374, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. 28, 147

Matusik, W., Pfister, H., Ngan, A., Beardsley, P.A., Ziegler, R. & McMillan, L. (2002). Image-based 3d photography using opacity hulls. *ACM Transactions on Graphics*, **21**, 427–437. 147

Matusik, W., Pfister, H., Brand, M. & McMillan, L. (2003). A data-driven reflectance model. *ACM Transactions on Graphics*, **22**, 759–769. 33

Meyer, M., Desbrun, M., Schröder, P. & Barr, A.H. (2002). Discrete differential-geometry operators for triangulated 2-manifolds. In *Proceedings of VisMath*, 35–57. 47

Moeslund, T., Hilton, A. & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, **104**, 90–126. 28

Murray, R.M., Sastry, S.S. & Zexiang, L. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edn. 22

Nehab, D., Rusinkiewicz, S., Davis, J. & Ramamoorthi, R. (2005). Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics*, **24**. 39, 67, 113

Ng, R., Ramamoorthi, R. & Hanrahan, P. (2004). Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics*, **23**, 477–487. 34, 164

Ngan, A., Durand, F. & Matusik, W. (2005). Experimental analysis of brdf models. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, EGSR'05, 117–126, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland. 15

Phong, B.T. (1975). Illumination for computer generated pictures. *Commun. ACM*, **18**, 311–317. 15

Pighin, F., Szeliski, R. & Salesin, D. (1999). Resynthesizing facial animation through 3d model-based tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 143–150. 31

Plankers, R. & Fua, P. (2001). Tracking and Modeling People in Video Sequences. *Computer Vision and Image Understanding*, **81**, 285–302. 29

Pons, J.P., Keriven, R. & Faugeras, O. (2005). Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 822–827 vol. 2. 26

Popa, T., South-Dickinson, I., Bradley, D., Sheffer, A. & Heidrich, W. (2010). Globally consistent space-time reconstruction. *Computer Graphics Forum (Proc. SGP)*. 31

Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, **108**. 28, 78

Ramamoorthi, R. (2005). Modeling illumination variation with spherical harmonics. *Face Processing: Advanced Modeling and Methods*. 12, 14

Ramamoorthi, R. & Hanrahan, P. (2001a). An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 497–500, ACM, New York, NY, USA. 34

Ramamoorthi, R. & Hanrahan, P. (2001b). On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A*, **18**, 2448–2459. 60

Ramamoorthi, R. & Hanrahan, P. (2001c). A signal-processing framework for inverse rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 117–128, ACM, New York, NY, USA. 11, 12, 34

Ramamoorthi, R. & Hanrahan, P. (2002). Frequency space environment map rendering. *ACM Transactions on Graphics*, **21**, 517–526. 127

Ramamoorthi, R. & Hanrahan, P. (2004). A signal-processing framework for reflection. *ACM Transactions on Graphics*, **23**, 1004–1042. 12, 16

Raskar, R., Nii, H., deDecker, B., Hashimoto, Y., Summet, J., Moore, D., Zhao, Y., Westhues, J., Dietz, P., Barnwell, J., Nayar, S., Inami, M., Bekaert, P., Noland, M., Branzoi, V. & Bruns, E. (2007). Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators. *ACM Transactions on Graphics*, **26**. 78

Reynolds, M., Doboš, J., Peel, L., Weyrich, T. & Brostow, G.J. (2011). Capturing time-of-flight data with confidence. In *IEEE Conference on Computer Vision and Pattern Recognition*. 52

Rother, C., Kolmogorov, V. & Blake, A. (2004). "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, **23**, 309–314. 128, 136

Rousseeuw, P. & Leroy, A. (1987). *Robust Regression and Outlier Detection*. John Wiley and Sons, New York. 65

Samaras, D., Metaxas, D., Ascalfua, P. & Leclerc, Y.G. (2000). Variable albedo surface reconstruction from stereo and shape from shading. In *IEEE Conference on Computer Vision and Pattern Recognition*, 480–487. 27

Sato, Y., Wheeler, M.D. & Ikeuchi, K. (1997). Object shape and reflectance modeling from observation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, 379–387, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. 33

Seitz, S., Curless, B., Diebel, J., Scharstein, D. & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 519–528. 25, 39

Shi, B., Matsushita, Y., Wei, Y., Xu, C. & Tan, P. (2010). Self-calibrating photometric stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1118–1125. 26

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. & Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, 1297–1304. 29

Sidenbladh, H., Black, M. & Fleet, D. (2000). Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, 702–718. 78

Sigal, L., Balan, A. & Black, M. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *Int'l Journal of Computer Vision*, **87**, 4–27. 28, 78

Sorkine, O. (2005). Laplacian mesh processing. In *STAR Proceedings of Eurographics 2005*, 53–70, Eurographics Association. 102, 106, 110

Starck, J. & Hilton, A. (2007). Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, **27(3)**, 21–31. 28, 78, 147

Stoll, C., Gall, J., de Aguiar, E., Thrun, S. & Theobalt, C. (2010). Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics*, **29**, 139:1–139:10. 147

Stoll, C., Hasler, N., Gall, J., Seidel, H.P. & Theobalt, C. (2011). Fast articulated motion tracking using a sums of gaussians body model. In *IEEE International Conference on Computer Vision*. 28, 78, 133

Sun, D., Roth, S., Lewis, J.P. & Black, M.J. (2008). Learning optical flow. In *European Conference on Computer Vision*, vol. 5304, 83–97. 20

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M. & Rother, C. (2008). A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 1068–1080. 89

Taylor, J., Shotton, J., Sharp, T. & Fitzgibbon, A.W. (2012). The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 103–110. 29

Tevs, A., Berner, A., Wand, M., Ihrke, I., Bokeloh, M., Kerber, J. & Seidel, H.P. (2012). Animation cartography: intrinsic reconstruction of shape and motion. *ACM Transactions on Graphics*, **31**, 12:1–12:15. 30

Theobalt, C., Ahmed, N., Lensch, H.P.A., Magnor, M.A. & Seidel, H.P. (2007). Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics*, **13**, 663–674. 33, 126, 147

Torrance, K.E. & Sparrow, E.M. (1967). Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.*, **57**, 1105–1112. 124

Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C. & Theobalt, C. (2010). Joint estimation of motion, structure and geometry from stereo sequences. In *European Conference on Computer Vision*, vol. 6314 of *Lecture Notes in Computer Science*, 568–581, Springer Berlin Heidelberg. 18, 19, 20, 21, 128

Valgaerts, L., Bruhn, A., Mainberger, M. & Weickert, J. (2012a). Dense versus sparse approaches for estimating the fundamental matrix. *International Journal of Computer Vision*, **96**, 212–234. 102

Valgaerts, L., Wu, C., Bruhn, A., Seidel, H.P. & Theobalt, C. (2012b). Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, **31**, 187:1–187:11. 5, 6, 7, 100

Vedula, S., Rander, P., Collins, R. & Kanade, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 475–480. 17

REFERENCES

Vlasic, D., Baran, I., Matusik, W. & Popović, J. (2008). Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*. 1, 3, 29, 59, 78, 147

Vlasic, D., Peers, P., Baran, I., Debevec, P., Popovic, J., Rusinkiewicz, S. & Matusik, W. (2009). Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics*, **28**, 174. 3, 28, 29, 123, 125

Vogiatzis, G. & Hernández, C. (2011). Self-calibrated, multi-spectral photometric stereo for 3D face capture. *Int'l Journal of Computer Vision*. 32, 99

Vogiatzis, G., Hernandez, C., Torr, P.H.S. & Cipolla, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 2241–2246. 26

Wand, M., Adams, B., Ovsjanikov, M., Berner, A., Bokeloh, M., Jenke, P., Guibas, L., Seidel, H.P. & Schilling, A. (2009). Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data. *ACM Transactions on Graphics*, **28**, 15:1–15:15. 31

Wang, Y., Huang, X., Su Lee, C., Zhang, S., Li, Z., Samaras, D., Metaxas, D., Elgammal, A. & Huang, P. (2004). High resolution acquisition, learning and transfer of dynamic 3-D facial expressions. *Computer Graphics Forum*, **23**, 677–686. 31

Waschbüsch, M., Würmlin, S., Cotting, D., Sadlo, F. & Gross, M. (2005). Scalable 3D video of dynamic scenes. In *Proc. Pacific Graphics*, 629–638. 28

Wei, X. & Chai, J. (2010). Videomocap: modeling physically realistic human motion from monocular video sequences. *ACM Transactions on Graphics*, **29**, 42:1–42:10. 28

Wei, X., Zhang, P. & Chai, J. (2012). Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics*, **31**, 188:1–188:12. 29

Weise, T., Leibe, B. & Gool, L.J.V. (2007). Fast 3D scanning with automatic motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 31

Weise, T., Bouaziz, S., Li, H. & Pauly, M. (2011). Realtime performance-based facial animation. *ACM Transactions on Graphics*, **30**, 77:1–77:10. 31

Williams, L. (1990). Performance-driven facial animation. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, 235–242, ACM, New York, NY, USA. 31

Wilson, C.A., Ghosh, A., Peers, P., Chiang, J.Y., Busch, J. & Debevec, P. (2010). Temporal upsampling of performance geometry using photometric alignment. *ACM Transactions on Graphics*, **29**, 17:1–17:11. 32, 60, 99

Woodham, R.J. (1980). Photometric method for determining surface orientation from multiple images. *Optical Engineering*, **19**, 139–144. 26

Wu, C., Liu, Y., Dai, Q. & Wilburn, B. (2010). Fusing multi-view and photometric stereo for 3d reconstruction under uncalibrated illumination. *IEEE Transactions on Visualization and Computer Graphics*. 27

Wu, C., Varanasi, K., Liu, Y., Seidel, H.P. & Theobalt, C. (2011a). Shading-based dynamic shape refinement from multi-view video under general illumination. In *IEEE International Conference on Computer Vision*. 5, 6, 7, 61

Wu, C., Wilburn, B., Matsushita, Y. & Theobalt, C. (2011b). High-quality shape from multi-view stereo and shading under general illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, 969–976. 5, 6, 7, 40

Wu, C., Varanasi, K. & Theobalt, C. (2012). Full body performance capture under uncontrolled and varying illumination: A shading-based approach. In *European Conference on Computer Vision*, 748–77. 5, 6, 7, 79

Wu, C., Stoll, C., Valgaerts, L. & Theobalt, C. (2013). On-set performance capture of multiple actors with a stereo camera. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, **32**, 161:1–161:11. 5, 7, 122

Yoon, K.J., Prados, E. & Sturm, P. (2010). Joint estimation of shape and reflectance using multiple images with known illumination conditions. *Int'l Journal of Computer Vision*, **86**, 192–210. 34

Yu, Y. & Malik, J. (1998). Recovering photometric properties of architectural scenes from photographs. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 207–217, ACM, New York, NY, USA. 33

Yu, Y., Debevec, P., Malik, J. & Hawkins, T. (1999). Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, 215–224, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. 33

Zhang, L., Noah, Curless, B. & Seitz, S.M. (2004). Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics*, **23**, 548–558. 31, 99

Zhang, R., Tsai, P., Cryer, J. & Shah, M. (1999). Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, 690–706. 26, 39, 127

Zimmer, H., Bruhn, A. & Weickert, J. (2011). Optic flow in harmony. *Int'l Journal of Computer Vision*, **93**, 368–388. 20, 118

Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S. & Szeliski, R. (2004). High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, **23**, 600–608. 28