



**Max-Planck-Institut für Informatik
Graphics-Optics-Vision Group
Saarbrücken, Germany**

BRDF Reconstruction from Video Streams of Multi-View Recordings

Master Thesis in Computer Science

Computer Science Department
University of Saarland

Naveed Ahmed

Supervisors: Dr. Hendrik Lensch
Dr. Marcus Magnor

Max-Planck-Institut für Informatik
Graphics-Optics-Vision Group
Saarbrücken, Germany

Begin: February 26, 2004
End: August 19, 2004

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Mastersarbeit selbständig und ohne fremde Hilfe verfasst habe. Ich habe dazu keine weiteren als die angeführten Hilfsmittel benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet.

Saarbrücken, den 19. August, 2004

Naveed Ahmed

Abstract

Synthesizing photorealistic images is an active area of research in computer graphics. Image based rendering combined with inverse rendering methods is used to generate photorealistic images from real world images under novel illumination conditions. Traditionally, very high-quality real world images of static objects, obtained under known viewing and lighting conditions are used in inverse rendering for the measurement of surface reflectance properties. This thesis focuses on surface material reconstruction of dynamic objects from video streams of multi-view recordings. Working with fairly low resolution movie streams of a dynamic object recorded in known viewing conditions and a geometry model tracked through all time steps, we approximate the best light source configuration, and measure the bidirectional reflectance distribution function of the object. We construct diffuse and specular maps for the whole sequence, and a diffuse correction map for each time step. We have applied our method to sequences of a human actor and are now able to synthesize views of the actor in arbitrary poses under arbitrary lighting conditions.

Acknowledgement

I want to thank Prof. Dr. Hans-Peter Seidel for all the support I received from the Computer Graphics group at MPI and my supervisors: Dr. Hendrik Lensch and Dr. Marcus Magnor for their help and guidance throughout this thesis work.

I thank all my colleagues of the Computer Graphics and the Graphics-Optics-Vision groups at MPI, in particular Ivo Ihrke, Gernot Ziegler and Edilson de Aguiar for helping me in reviewing the text and Akiko Yoshida for helping with \LaTeX . I am grateful to some of my very good friends: Hang Yu for all his help during this work, Imran Rauf and Shah Jamal Alam who made this work and my life at Saarbrücken much easier.

Finally, I am indebted to my family for their continuous support during my studies. In spite of the distance they were an incessant source of encouragement. My gratitude for Abbu, Ammi, Shoby, Fariha and Faiza is beyond words.

Contents

| | |
|---|------------|
| Abstract | i |
| Acknowledgement | iii |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Goal | 2 |
| 1.3 Thesis Outline | 3 |
| 2 Background | 5 |
| 2.1 Light Transport in a Scene | 5 |
| 2.2 Radiometry | 6 |
| 2.3 Radiometric Quantities | 6 |
| 2.4 The Rendering Equation | 7 |
| 2.5 Bidirectional Reflectance Distribution Function | 8 |
| 2.5.1 BRDF Properties | 9 |
| 2.6 Reflectance | 10 |
| 2.7 BRDF Representation | 11 |
| 2.7.1 Tabulated BRDF | 11 |
| 2.7.2 Spherical Harmonics | 11 |
| 2.7.3 BRDF models | 11 |
| 3 Related Work | 15 |
| 3.1 Inverse Rendering | 15 |
| 3.1.1 Inverse Lighting | 16 |

| | | |
|----------|--|-----------|
| 3.1.2 | Inverse Geometry | 16 |
| 3.1.3 | Inverse Reflectance | 17 |
| 3.2 | Free-Viewpoint Video of Human Actors | 17 |
| 3.2.1 | Overview | 17 |
| 3.2.2 | Multi-View Video Recording | 18 |
| 3.2.3 | Human Body Model | 19 |
| 3.3 | Discussion | 20 |
| 4 | Modus Operandi | 21 |
| 4.1 | Algorithm | 21 |
| 4.1.1 | Light Source Estimation | 21 |
| 4.1.2 | BRDF Estimation | 22 |
| 4.2 | Rendering | 22 |
| 5 | Texture Resampling | 25 |
| 5.1 | Texture Atlas | 25 |
| 5.1.1 | Hardware-Accelerated Resampling | 28 |
| 5.1.2 | Grown Texture Atlas | 30 |
| 5.1.3 | Averaged Texture Atlas | 30 |
| 5.2 | Discussion | 31 |
| 6 | BRDF Reconstruction | 33 |
| 6.1 | Light Source Estimation | 33 |
| 6.1.1 | Assembling Reflectance Values | 35 |
| 6.1.2 | BRDF Estimation | 36 |
| 6.1.3 | Error Estimation | 38 |
| 6.1.4 | Best Light Source Configuration | 39 |
| 6.2 | Reflectance Measurement | 41 |
| 6.2.1 | Clustering | 41 |
| 6.2.2 | Assembling Samples | 42 |
| 6.2.3 | BRDF Fitting | 43 |
| 6.3 | Discussion | 44 |

| | |
|---|-----------|
| <i>CONTENTS</i> | vii |
| 7 Results | 45 |
| 7.1 Rendering | 45 |
| 7.1.1 Assembling the BRDF atlases | 45 |
| 7.1.2 Correction atlas | 47 |
| 7.2 Discussion | 50 |
| 8 Conclusions & Future Work | 53 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Radiometric Quantities | 6 |
| 6.1 | Average and diffuse errors for light source configurations. | 40 |
| 6.2 | Comparison of the two optimization methods. | 43 |
| 7.1 | Average error for the BRDF and the correction atlas rendering. . . | 49 |

List of Figures

| | | |
|-----|--|----|
| 2.1 | Re-Parameterization of the rendering equation. | 8 |
| 2.2 | Model description for interaction of light with matter | 9 |
| 2.3 | Classification of the surface reflectance. | 10 |
| 2.4 | Cosine lobe with respect to different exponent parameters. | 13 |
| 3.1 | Multi-view video recording | 18 |
| 3.2 | The human body model | 19 |
| 5.1 | Projective texture mapping | 26 |
| 5.2 | Surface parameterization into the texture atlas. | 27 |
| 5.3 | Resampling into the texture atlas. | 28 |
| 5.4 | Growing texture atlas | 29 |
| 5.5 | Averaged texture atlas. | 30 |
| 6.1 | Light source distribution. | 34 |
| 6.2 | Pseudocode for the approximation of the best light source configuration. | 35 |
| 6.3 | Resampling the shadow atlas from the center light source. | 37 |
| 6.4 | Average BRDF estimation. | 39 |
| 6.5 | Diffuse BRDF estimation. | 40 |
| 6.6 | Clustering texels | 42 |
| 7.1 | Assembling the BRDF Atlases. | 46 |
| 7.2 | The model rendered with the BRDF atlases. | 46 |
| 7.3 | Pseudocode for generating the correction atlas. | 47 |
| 7.4 | The rendered diffuse atlas. | 48 |

| | | |
|-----|--|----|
| 7.5 | Comparison between the original and the reconstructed image. . . | 49 |
| 7.6 | The model rendered using the BRDF atlases. | 50 |
| 7.7 | The model rendered under changing illumination conditions. A single light source moves around the model. The camera changes after every 7th frame. | 51 |
| 7.8 | The model rendered from novel viewpoints using the correction atlases. | 52 |

Chapter 1

Introduction

1.1 Motivation

Producing photorealistic images is one of the major areas of focus in computer graphics. Two general methods are used to render photorealistic images: ray tracing, and (image based) rendering. Image based rendering approaches range from pure non-geometrical techniques like Light Field Rendering [9], to hybrid approaches like Surface Light Fields [26]. In ray tracing, model accuracy plays an important role in generating photorealistic images. A high quality model with accurate illumination and reflection properties results in high quality synthesized images. An image can be rendered given a known camera position, illumination conditions, model details, and model reflection properties. Inverse rendering can be used to determine the attributes of the scene [13]. With known model details, light source information, and recorded images, one can solve for surface reflection properties by measuring the bidirectional reflectance distribution function (BRDF). Measured BRDFs can be used to render the model from novel views and illumination conditions, while maintaining realistic appearance at low storage cost. Marschner [13] presents three inverse rendering problems to solve for two of the attributes; inverse lighting to solve for illumination, photographic texture measurement and image based BRDF measurement to solve for reflectance. Lensch et al. [8] propose an approach to capture spatially varying BRDFs. BRDF measurements require multiple images of the object from different known view-

ing directions and lighting conditions. Traditionally, very high quality images of static objects obtained from digital cameras are used for this purpose.

Our approach in this thesis is to measure BRDFs from multi-view video recordings. We work on video streams of dynamic objects recorded from different cameras. Video streams are recorded at fairly low resolution and marred by temporal aliasing. We try to exploit the availability of large amounts of data to get accurate measurement. We use a combination of photographic texture measurement [13] and image-based BRDF measurement [8] algorithms to approximate the best light source configuration, and to reconstruct surface reflectance properties of the object.

1.2 Goal

As mentioned in the previous section, the main goal of this work is to measure BRDFs from multi-view video recordings. We work on existing data from the previous work on Free-Viewpoint Video of Human Actors [2]. As input we have low resolution (320 x 240 pixels) synchronized multi-view video streams of the dynamic object recorded at 15 frames per second from eight different cameras. Viewing conditions are known, while lighting conditions are partially known as we do not have exact positions of light sources but only their direction and layout.

We intend to capture separate BRDFs for each point on our object. Since we only work with the partial light source information, reliable fitting of the specular part of the BRDF is difficult because of its dependency on spatial attributes of the light source. We use the image-based BRDF measurement algorithm to best fit the specular component for individual clusters of similar materials and an algorithm similar to photographic texture measurement [13] for the diffuse component, while approximating the best light source configuration that results in minimum error for our measured BRDF.

1.3 Thesis Outline

Chapter 2 gives a brief overview of necessary background material, while related work is categorized and discussed in Chapter 3.

Overview of our algorithm is presented in Chapter 4. Chapter 5 presents the resampling process in detail. Motion parameters, transformation matrices, and texture parameterization are used to resample input textures into texture atlases. This resampling is done on graphics hardware.

Chapter 6 explains the process of BRDF reconstruction. Algorithms for estimating the best light source configuration, assembling per-triangle and per-textel reflectance values, clustering, and BRDF estimation are presented. The correction atlas for each time step, hardware accelerated rendering and results of our work are presented in Chapter 7. We conclude the thesis by discussing our results and future work in Chapter 8.

Chapter 2

Background

In this chapter we will review fundamental principles behind light transport in a scene and interaction of light with matter. These principles will be extended to derive the rendering equation and the bidirectional reflectance distribution function.

2.1 Light Transport in a Scene

Light is electromagnetic radiation with a wavelength that is visible to the eye [25]. It exhibits dual properties of wave and particle. Particle theory describes light traveling in discrete energy packets called photons and explains interaction of light with surfaces, while wave theory describes it as an electromagnetic wave and explains phenomena such as interference, polarization, and diffraction.

Light propagation in a scene can be simulated by forward and backward light transport. In nature, forward light transport starts with photons being emitted from some light source, they hit some surface and undergo reflection, or refraction, while losing some energy, until they are absorbed. Forward light transport is a parallel process at the speed of light. Backward light transport is far less expensive to simulate. Ray tracing starts with rays from the camera and traces them back to the light source thus taking only contributing rays into account.

2.2 Radiometry

Light transport in computer graphics is usually simulated in terms of rays, where rays of some specific energy travel in some specific direction. Interaction of rays with one other is ignored, thus resulting in loss of the phenomena due to the wave nature of light. Interaction of a ray with a surface is handled to approximate particle theory. This results in energy transfer. Radiometry is the discipline of measuring radiant energy transfer. Radiometric quantities have physical meaning and can be measured accurately with proper equipment.

2.3 Radiometric Quantities

In this section we define some of the most commonly used radiometric quantities [14].

| Quantity | Symbol | Unit | Definition |
|----------------|--------|-----------------------------|--|
| Radiant energy | Q | <i>joule(J)</i> | The total energy in a radiation field or total energy delivered by such a radiation field. |
| Radiant flux | Φ | <i>watt(W)</i> | The rate at which radiant energy is transferred from one region to another by the radiation field $\Phi = dQ / dt$, where t is time. |
| Radiance | L | <i>watt/m²sr</i> | Total flux traveling in space at some point x in some specific direction, per unit area perpendicular to the direction of travel, per solid angle. |
| Irradiance | E | <i>watt/m²</i> | Total flux per unit area incident onto a surface from some specific direction. |
| Radiosity | B | <i>watt/m²</i> | Total flux per unit area leaving a surface in some specific direction. |

Table 2.1: Radiometric Quantities

2.4 The Rendering Equation

Energy equilibrium in a scene is expressed by the Rendering Equation. Kajiya [4] presented the rendering equation as a generalization of image synthesis algorithms:

$$L_o(\vec{x}, \vec{\omega}_o) = L_e(\vec{x}, \vec{\omega}_o) + \int_{\Omega_+} f_r(\vec{\omega}_i, \vec{x}, \vec{\omega}_o) L_i(\vec{x}, \vec{\omega}_i) \cos\theta_i d\omega_i \quad (2.1)$$

The outgoing illumination L_o , at position \vec{x} and outgoing direction $\vec{\omega}_o$ is equal to the sum of self-emission L_e at this position and incoming radiance L_i from all incoming directions $\vec{\omega}_i$, weighted by the direction dependent reflectance $f_r(\vec{\omega}_i, \vec{x}, \vec{\omega}_o)$. Eq. (2.1) can be re-parameterized over all scene surfaces S :

$$L_o(\vec{x}, \vec{\omega}_o) = L_e(\vec{x}, \vec{\omega}_o) + \int_{\vec{y} \in S} f_r(\vec{\omega}_i, \vec{x}, \vec{\omega}_o) L_o(\vec{y}, -\vec{\omega}_i(\vec{x}, \vec{y})) G(\vec{x}, \vec{y}) dA_y \quad (2.2)$$

The integral goes over all scene surfaces S , $L_o(\vec{y}, -\vec{\omega}_i(\vec{x}, \vec{y}))$ is the outgoing radiance of some surface at position \vec{y} in the direction of \vec{x} . $G(\vec{x}, \vec{y})$ is the geometry term that takes into account geometrical properties of radiance transfer between the surfaces \vec{x} and \vec{y} (Figure 2.1):

$$G(\vec{x}, \vec{y}) = V(\vec{x}, \vec{y}) \frac{\cos\theta_x \cos\theta_y}{\|\vec{x} - \vec{y}\|^2} dA_y \quad (2.3)$$

$V(\vec{x}, \vec{y})$ is the visibility term between \vec{x} and \vec{y} . It is 1 if \vec{x} is visible from \vec{y} and 0 otherwise.

Eq. (2.2) can be solved to obtain a generalized global illumination solution. In computer graphics some approximations are used to obtain computationally feasible algorithms. Instead of the full spectrum of visible wavelengths, the rendering equation is evaluated for red, green, and blue only. Surfaces are sampled as small patches assuming locally constant reflection and visibility. The hemisphere is sampled along discrete finite directions, reducing the integral to a summation. Similarly a simplified reflection function model is used instead of measured data. The reflection model will be discussed in detail in the next section.

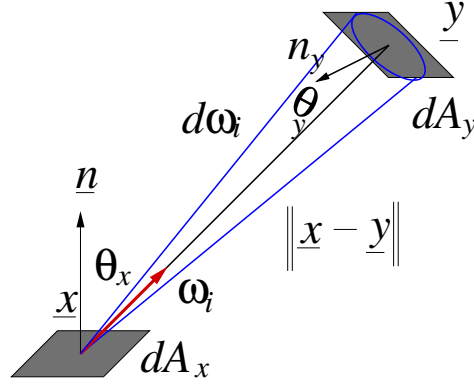


Figure 2.1: Re-Parameterization of the rendering equation.

2.5 Bidirectional Reflectance Distribution Function

Reflectance properties of a surface can be represented by the so-called bidirectional reflectance distribution function (BRDF) that describes the interaction of light with matter. A hierarchy of models describing the interaction of light with matter can be observed in Figure 2.2 [7]. As can be seen in Figure 2.2, the spatially varying BRDF is a 6D function and is one of the two simplifications of the bidirectional subsurface scattering distribution function (BSSRDF). Subsurface scattering describes the phenomenon where light enters an object at one point and leaves from some other point. Such materials can be translucent and transparent objects. In this thesis, we do not consider subsurface scattering, only opaque surfaces and their BRDF.

The BRDF describes surface reflectance for light incident from direction (θ_i, φ_i) as observed from direction (θ_o, φ_o) . It is bidirectional because it depends on two directions and the surface position (6D function):

$$f_r(\vec{\omega}_o, \vec{x}, \vec{\omega}_i) = \frac{L_o(\vec{x}, \vec{\omega}_o)}{L_i(\vec{x}, \vec{\omega}_i) \cos \theta_i d\vec{\omega}_i} \quad (2.4)$$

The BRDF is defined as the ratio of outgoing radiance L_o at position \vec{x} in direction $\vec{\omega}_o$, to incoming irradiance L_i arriving at position \vec{x} from direction $\vec{\omega}_i$. It has the unit sr^{-1} .

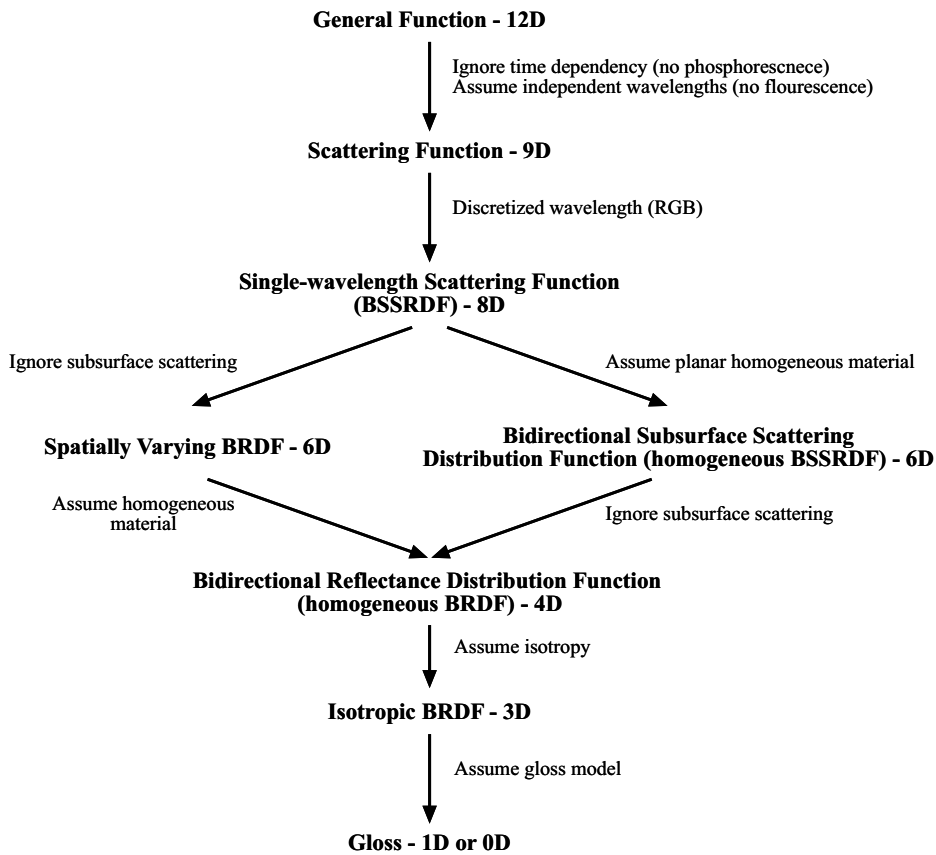


Figure 2.2: Model description for interaction of light with matter. General description starts with a 12D function, at each level certain assumptions result in a reduction in dimensionality.

2.5.1 BRDF Properties

- BRDF obeys the Helmholtz reciprocity principle, i.e. the BRDF remains unchanged if incident and reflection directions are interchanged.
- For isotropic surfaces where reflectivity is independent of rotation around the surface normal, the isotropic BRDF has only 3 instead of 4 degrees of freedom $f_r(\vec{x}, \theta_i, \theta_o, \varphi_o - \varphi_i)$.
- Real world materials never emit light but absorb some of the energy thus the BRDF is energy conserving.

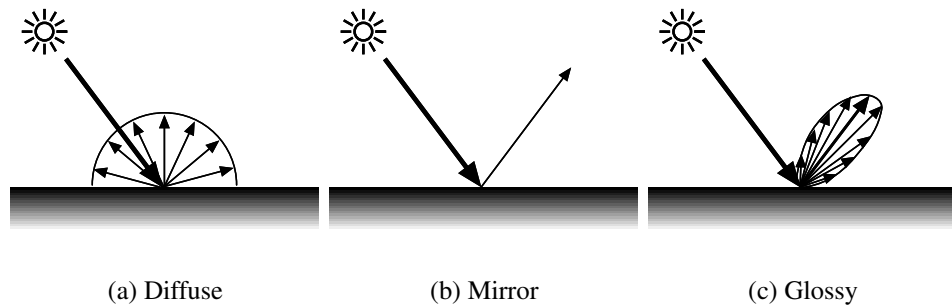


Figure 2.3: Classification of the surface reflectance.

2.6 Reflectance

We can represent reflectance properties of objects with the BRDF. Observed reflectance depends on the properties of the materials together with viewing and illumination conditions. Reflectance may vary due to

- Illumination angle
- Viewing angle
- Wavelength of light

in addition to surface properties such as

- Surface micro-geometry
- Absorption properties
- Index of refraction
- Subsurface Scattering

Reflectance can be classified into three categories. The BRDF can be considered to be composed of all three types, reflectance categories are shown in Figure 2.3:

- Diffuse Reflection: This is a view independent reflectance and also called Lambertian [6] reflectance. In diffuse reflectance the incident light is reflected equally into all directions. The surface looks exactly the same from all viewing angles.

- **Specular Reflection:** This view dependent reflection can be observed in mirrors and other highly polished surfaces. Part of the incident light source is reflected in a single direction.
- **Glossy Reflection:** Due to the roughness of the surface light may reflect around the specular direction instead of reflecting in the perfect mirror direction. The size of spherical lobe around the reflection vector (Figure 2.3(c)) is determined by the roughness of the surface.

2.7 BRDF Representation

Reflectance properties of materials are represented by the BRDF. A single BRDF model with varying parameters can be used to represent a subset of materials. Alternatively each material can be represented by its own BRDF. BRDF representations range from physics-based models to pure empirical models that can result in realistic appearance.

2.7.1 Tabulated BRDF

Measured [24] or simulated [3] BRDF data can be stored in tabular form and then be interpolated for novel incident and outgoing directions. The storage cost for the tabulated BRDFs is very high, usually a more compact representation is used by fitting a model on tabulated data.

2.7.2 Spherical Harmonics

BRDF is represented as a sum of scaled basis functions. The measured BRDF data is approximated by spherical harmonic functions and by means of computationally inexpensive methods, it can be reconstructed.

2.7.3 BRDF models

BRDF models are used to evaluate the reflectance for incoming and outgoing directions using only a small set of parameters. The parameters differ for each ma-

terial and can be determined by approximating the appearance of the material or by fitting a model to available data. BRDF models fall into two categories, namely physics-based and empirical models. The Lambertian reflectance model is relatively simplistic as light is reflected evenly in all directions. The main difference between different models lies in their approximation of the specular component.

Physics-based models

Physics-based models represent the specular component of the BRDF by assuming a distribution model for surface microfacets. The Cook-Torrance [22] model assumes a Gaussian distribution for the microfacets. It also incorporates a Fresnel term and a Geometry term for shadowing and masking of microfacets.

Empirical Models

Physics-based models are best suited to represent surface reflectance, but due to intricate details they are complex to evaluate; therefore various empirical models are proposed to approximate the BRDF. Phong [15] proposed one of the earlier empirical models. We use this model to fit it to our data. One of the drawbacks of the Phong model is that it is not energy conserving. The Phong model is described by the following equation:

$$f(\vec{l}, \vec{v}, \vec{x}) = k_d(\vec{l} \cdot \vec{n}) + k_s(R(\vec{l}) \cdot \vec{v})^{k_e} \quad (2.5)$$

For the light source position \vec{L} , the viewing position \vec{V} and the surface position \vec{x} , the light vector \vec{l} is defined as:

$$\vec{l} = \vec{L} - \vec{x} \quad (2.6)$$

Similarly the viewing vector \vec{v} is given by:

$$\vec{v} = \vec{V} - \vec{x} \quad (2.7)$$

\vec{n} is the normal vector at \vec{x} , and $R(\vec{l})$ is the reflection vector defined as:

$$R(\vec{l}) = \vec{l} - 2(\vec{l} \cdot \vec{n})\vec{n} \quad (2.8)$$

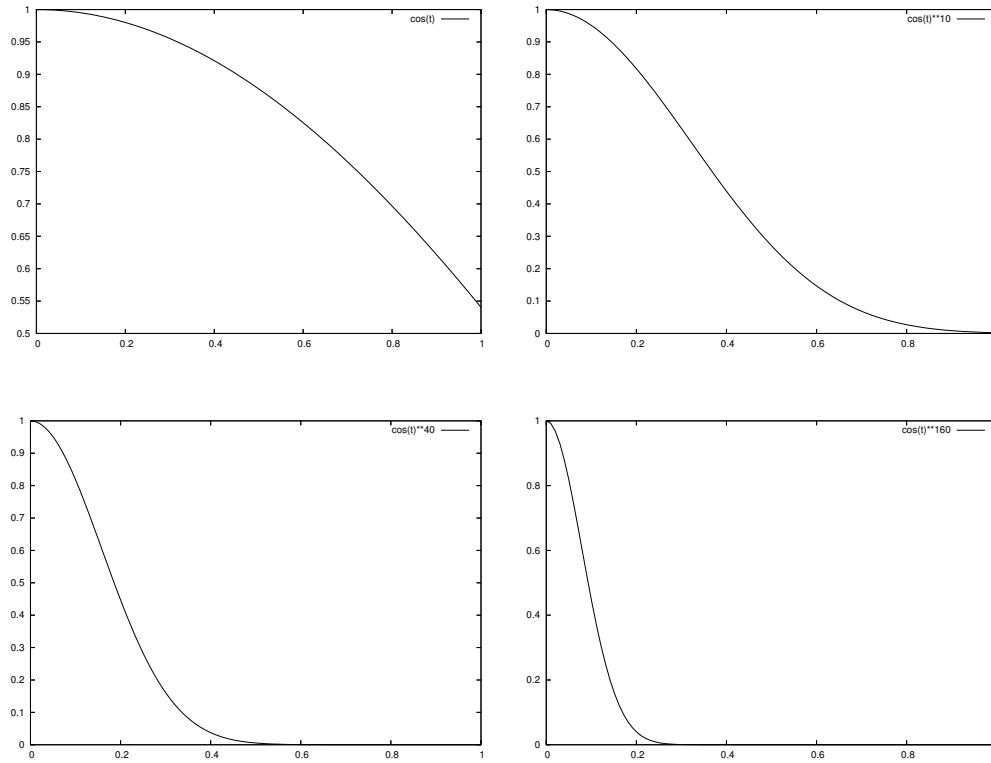


Figure 2.4: Cosine lobe with respect to different exponent parameters.

The three parameters k_d , k_s , and k_e are the diffuse and specular components, and the shininess exponent respectively. The size of the highlight is determined by the so-called Phong Exponent k_e . A higher exponent results in a sharper highlight as can be seen in Figure 2.4. The Phong model (Eq. (2.5)) can be extended to incorporate more than one light source. For k light vectors $\vec{l}_1, \vec{l}_2, \dots, \vec{l}_k$, viewing vector \vec{v} and position \vec{x} , Eq. (2.9) shows an extended Phong model, where f_r is the resulting reflectance:

$$f_r = k_d \sum_i^k (\vec{l}_i \cdot \vec{n}) + k_s \sum_i^k (R(\vec{l}_i) \cdot \vec{v})^{k_e} \quad (2.9)$$

We fit this model to our data. Others model could have been used as well. We chose this model because of its simplicity in terms of parameters and evaluation. In Chapter 6, we approximate the illumination conditions by evaluating different

configurations of point light sources. The estimation of the light source configuration involves the resampling of input data and model fitting. Since we deal with large amounts of data, this model with only three parameters and its evaluation that requires linear time with respect to the number of light sources, is well suited for our work.

Chapter 3

Related Work

In this chapter we categorize and discuss some of the work related to inverse rendering. We will also review the Free-Viewpoint Video of Human Actors [2] project, as we use the data from that system for BRDF reconstruction.

3.1 Inverse Rendering

Inverse Rendering has been an active area of research in recent years. In Chapter 1 we briefly discussed some inverse rendering problems, we can formalize it by inverting the rendering equation:

$$L_e(\vec{x}, \vec{\omega}_o) = L_o(\vec{x}, \vec{\omega}_o) - \int_{\Omega_+} f_r(\vec{\omega}_i, \vec{x}, \vec{\omega}_o) L_i(\vec{x}, \vec{\omega}_i) \cos\theta_i d\omega_i \quad (3.1)$$

For some known observed illumination $L_o(\vec{x}, \vec{\omega}_o)$, we can derive one of the scene attributes \vec{x} , f_r , or L_i . The observed illumination consists of a number of images generated by the scene attributes. The scene attributes are approximated by formulating the inversion as a minimization problem. The values obtained by solving the minimization problem are closest to the original with respect to some error measure usually based on the difference between recorded and reconstructed views of the scene. Marschner [13] presents three problems in this domain, solving for two of the scene attributes, lighting and reflectance. Traditionally, methods are proposed to solve for only one unknown scene attribute. Lensch et al. [8] use a

marker based approach to solve for the position of the point light source and then use image based BRDF measurement to solve for the reflectance. Ramamoorthi [17] presents a signal processing framework to solve for the lighting and the BRDF. In general, inverse rendering can be divided into three categories, inverse lighting, inverse geometry and inverse reflectance.

3.1.1 Inverse Lighting

Inverse lighting deals with recovering the attributes of the incident light L_i . Kawai et al. [5] present methods to recover position and intensity of incident lights. Marschner [13] recovers incident light fields under the assumptions that the environment is at infinity and no light is emitted or reflected into the environment from the object. Effects of global illumination or interreflections are also ignored. Most of the work in inverse lighting is limited to scenes obtained under strict illumination conditions, and are usually restricted to point light sources. The signal processing framework presented by Ramamoorthi [17] assumes general illumination conditions. It represents the BRDF with spherical harmonics as basis functions and the reflected light field is a convolution between the BRDF and the incoming light.

3.1.2 Inverse Geometry

Inverse geometry deals with the reconstruction of the scene geometry \vec{x} from a set of images. In case of a single image, it is known as shape from shading. A multi-image version of shape from shading is referred as photometric stereo [27]. Two or more images obtained under different illumination conditions are used to reconstruct the geometry, increasing the accuracy of the result. It is a very active field of research in computer vision for the last two decades. Silver [20] uses images of a wooden sphere as an empirical model of the material to reconstruct different wooden shapes under same lighting conditions. Basri et al. [1] present photometric stereo under general illumination conditions and Magda et al. [12] present an approach to reconstruct surfaces with arbitrary BRDFs.

3.1.3 Inverse Reflectance

Inverse reflectance deals with the reconstruction of surface reflectance f_r of an object from recorded images. The object can be composed of a single or multiple materials. In the latter case, the average BRDF for the whole object or individual BRDFs for each material are reconstructed. Marschner [13] presents work on two problems for unknown reflectance, photographic texture measurement and image based BRDF measurement. Photographic texture measurement solves for spatial variation in the object reflectance, constructing high resolution textures for rendering, while image based BRDF measurement captures reflectance by measuring the BRDF of a surface. Lensch et al. [8] use image based BRDF measurement to capture spatial variation in surface reflectance by representing the BRDF of each material as a linear combination of basis BRDFs.

3.2 Free-Viewpoint Video of Human Actors

BRDF reconstruction requires as input data a geometric model and recorded images of the object under different illumination. In this work we use a human body model and multi-video recordings, the data is provided by the Free-Viewpoint Video of Human Actors [2] project. We will review only the relevant parts of this project, primarily environment, method of acquisition and the quality of the input data.

3.2.1 Overview

In free-viewpoint video, the viewer can interactively choose his viewpoint in 3D space to observe the action of a dynamic real world scene from arbitrary perspectives [2]. It uses synchronized multi-view video streams of a human actor's performance to estimate the motion parameters of a generic body model and allows to render it interactively from any viewpoint. From the input data of synchronized multi-view video streams, silhouette images of the person are extracted in each camera view through background subtraction. An offline optimization process that minimizes the difference between the silhouettes and the rendered geometric

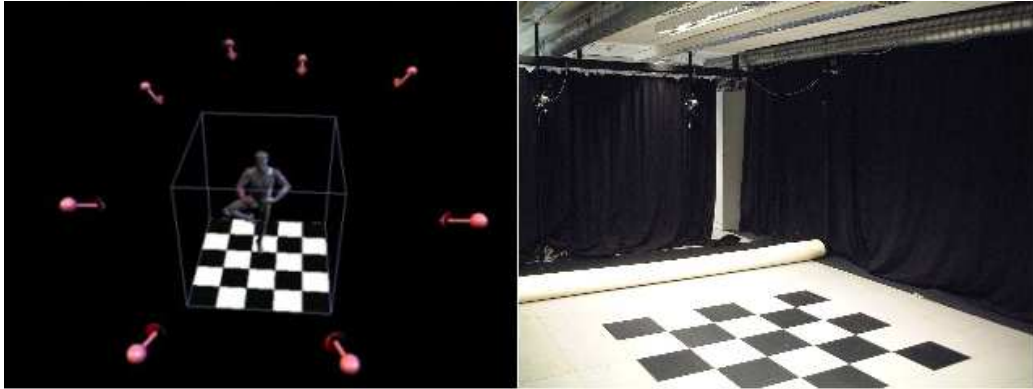


Figure 3.1: The image on the left shows the camera setup. Spheres represent the camera positions, the arrows depict camera orientation. The image on the right shows the studio.

model in different poses is used to determine the motion parameters. Rendering from an arbitrary viewpoint is achieved by generating a time dependent texture map from multi-view image data.

3.2.2 Multi-View Video Recording

The input video sequences are recorded in a multi-view camera studio (Figure 3.1). Eight IEEE1394 cameras are placed around the center of a room at approximately equal distance and angle, all of them are synchronized through an external trigger. A pair of cameras is controlled by an Athlon 1GHz PC that streams the recorded data directly to the disk. Video streams are recorded at a resolution of 320x240 pixels at 15 frames per second. The calibration of the cameras is determined using Tsai's algorithm [23]. After the calibration all cameras are registered in a global coordinate system. The studio walls are covered with black curtains to minimize the influence of external light sources. The studio is lit by multiple light sources at the ceiling of the room, forming a rectangular area with respect to the center of the room.



Figure 3.2: The surface model with underlying skeleton. Spheres indicate joints and the different parameterization for each. Green sphere - 1DOF, blue sphere - 3 DOF, and red sphere - 4DOF. The blue-black sphere at the pelvis represents 3 joints simultaneously, one for the root and the other two to connect the upper and lower parts of the model.

3.2.3 Human Body Model

The body model used in this project is a generic geometric model consisting of 16 hierarchical body segments (head, torso, upper arm, lower arm etc.). An underlying skeleton consisting of 17 joints that connect the single body segments is used to control the model's kinematics. Different parameterizations are used for different joints. The pelvis, being the root of the model, provides the degrees of freedom for global rotation and translation. Subsequent joints are parameterized in local coordinate systems with respect to their parent joints. 35 parameters are needed to completely define a body pose. A 1D Bezier Spline is defined along each segment in local coordinates to define non-uniform scaling which accommodates for segment deformation. Figure 3.2 shows the surface model (29965

triangles) with its underlying skeleton.

3.3 Discussion

We have discussed the acquisition environment and the method that generates the input data for this work. The input consists of a generic geometric body model (29965 triangles) and eight synchronized video streams. Each video stream has 398 frames recorded at a resolution of 320x240 pixels at 15 frames per second. The resolution of the images is low and the temporal resolution of 15 frames per seconds introduces temporal aliasing in the video streams. Viewing conditions are completely known, whereas only partial information about lighting conditions is available.

The acquisition is based on a generic geometric model. This significantly affects the reliability of acquired reflectance samples especially after texture-to-geometry registration. The human actor wears clothes whereas the model is naked. For some regions this results in the surface normals pointing in wrong directions.

We also reviewed inverse rendering and briefly discussed some problems in that domain. In this work we concentrate on reconstructing surface reflectance properties and approximate lighting conditions. We use a combination of approaches similar to photographic texture measurement [13] and image based BRDF measurement [8]. The former is used to construct BRDF texture maps, and the latter to approximate a BRDF model and the light source configuration.

Chapter 4

Modus Operandi

In this chapter we will present an overview of our algorithm and the rendering process. The acquisition process and the input data is explained in Chapter 3. Input data consists of a generic geometric body model and eight synchronized video streams. The model is animated and goes through geometrical transformations over time. To access the reflectance values of each surface point independent of its geometrical transformations, a texture atlas is defined that maps the 3D geometry into the 2D texture domain. A texture atlas for each body part is constructed and then stitched together to form a single texture. The texture atlas parameterization and resampling is discussed in detail in Chapter 5.

4.1 Algorithm

The estimation process consists of two parts. In the first part, the light source configuration is estimated while the BRDF estimation is done in the second part.

4.1.1 Light Source Estimation

The light source estimation algorithm is based on per-triangle BRDF estimation. A single reflectance value for each triangle is used. All the reflectance values in a triangle are averaged, which are then used to construct an averaged texture atlas. The procedure of constructing the averaged texture atlas is discussed in Section 5.1.3. We want to approximate the illumination conditions with a configuration

of point light sources distributed in a rectangular area at the ceiling of the room. Different configurations of light sources are considered. For each configuration, the following scheme is used:

- Estimate average k_d , k_s , and k_e for the whole model.
- Calculate average error for the estimated average BRDF.
- Subtract the specular contribution from all reflectance values to obtain per-triangle diffuse component k_d .
- Calculate the average error for the estimated per-triangle k_d .

The light source configuration with the minimum error is selected as the best configuration. The light source estimation process is discussed in detail in Section 6.1.

4.1.2 BRDF Estimation

The estimated best light source configuration is used in the BRDF estimation process. The BRDF estimation is performed for each texel in the texture atlas. For the estimation of the specular components of the BRDF (k_s , k_e) all reflectance values are classified into three clusters (Section 6.2.1). For each cluster, k_d , k_s , and k_e are estimated. Using the estimated k_s and k_e , the specular contribution is subtracted from all reflectance values to obtain per-texel diffuse component k_d . The BRDF estimation process is discussed in detail in Section 6.2.

4.2 Rendering

The texture atlas parameterization is used to construct texture atlases for each of the three BRDF parameters k_d , k_s , and k_e . A diffuse atlas is constructed from the estimated per-texel diffuse component k_d . The estimated BRDF for each cluster is used to construct k_s and k_e atlases.

To render the model realistically under the best light source configuration, a diffuse correction atlas for each time step is constructed. The correction atlas

stores the difference of the k_d atlas and the diffuse component of the corresponding input frame. The rendering process and the construction of the BRDF and the correction atlases are presented in Chapter 7.

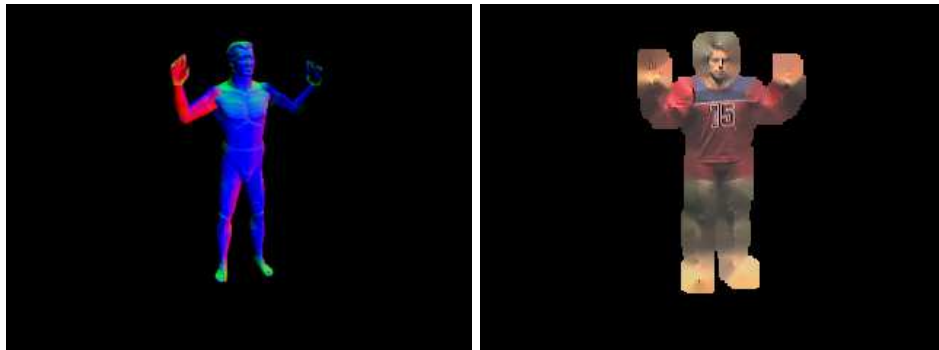
Chapter 5

Texture Resampling

As discussed in Chapter 3, we work on video streams of multi-view recordings from the Free-Viewpoint Video of Human Actors (Section 3.2) project. Using intrinsic and extrinsic parameters (Section 3.2.2), we calculate the modelview and projection matrices for each camera. Given the animation parameters i.e. transformation matrices for each body part and transformation matrices for each camera, we use projective texture mapping [19] to map a single texture at time t for a specific camera onto the model (Figure 5.1). The model exhibits the correct body pose for time t by applying motion parameters on the triangle mesh with the incorporated Free-Viewpoint Video of Human Actors framework. Figure 5.1(a) shows the initial pose of the surface model at time 0, Figure 5.1(b) shows the corresponding texture, while Figure 5.1(c) shows the resultant model after projective texture mapping. Projective texture mapping is performed in vertex and fragment shaders supported on consumer level graphics hardware. Because we rely on OpenGL texture lookup that uses filtering for the texture mapping, the texture frame in Figure 5.1(b) is grown outwards at the edges to eliminate artifacts that can arise when resampling triangles at the edges.

5.1 Texture Atlas

The mapping from input streams to texture space is done by defining a texture atlas which maps the surface of the 3D model into the 2D domain. The problem



(a) Surface model at time 0.

(b) Texture frame at time 0.



(c) Texture mapped model.

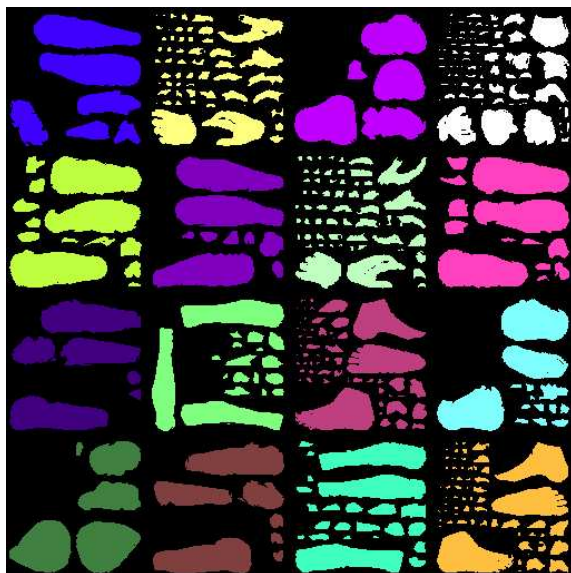
Figure 5.1: Using projective texture mapping to apply texture onto model for specific time and camera.

of creating a texture atlas is closely related to the problem of surface parameterization [7]. An arbitrary mesh can be parameterized onto a texture atlas by minimizing the texture stretch and deviation [18]. Marschner [13] and Sorkine [21] also proposed methods for creating the texture atlas. For general meshes it is necessary to introduce cuts on the surface in order to bound the distortion. Such cuts may partition the surface into distinct patches.

Our texture atlas is constructed by projecting the triangles of one patch orthogonally onto a plane defined by the average surface normal [28]. The selection of patches ensures a minimum sampling density of the surface, i.e. starting with one arbitrary seed triangle, neighboring triangles are added to the patch until the



(a) Color coded body parts.



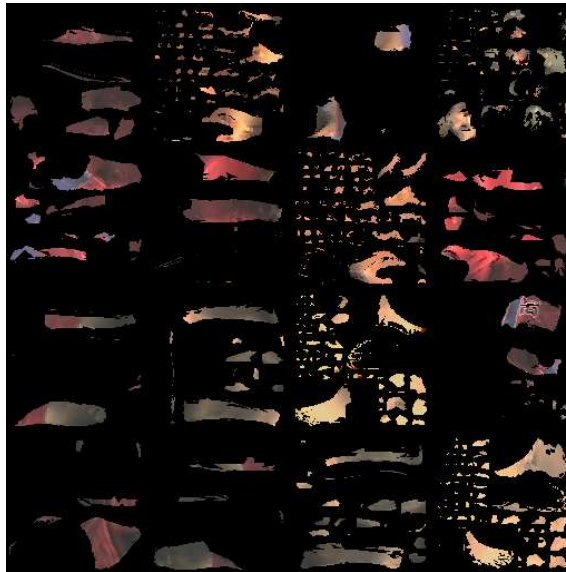
(b) Corresponding regions in texture space.

Figure 5.2: Surface parameterization into the texture atlas.

triangle normal deviates too much from the average normal. If one patch cannot grow any further another seed triangle starts a new patch. A separate texture atlas is constructed for each body part which are then joined into a single texture as demonstrated in Figure 5.2(a) and 5.2(b).



(a) Original frame (320x240) from arbitrary viewpoint.

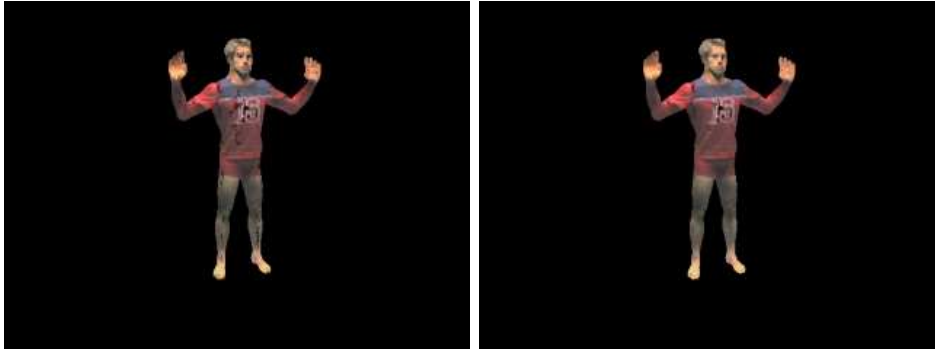


(b) Resampled frame (512x512) considering visibility.

Figure 5.3: Resampling into the texture atlas.

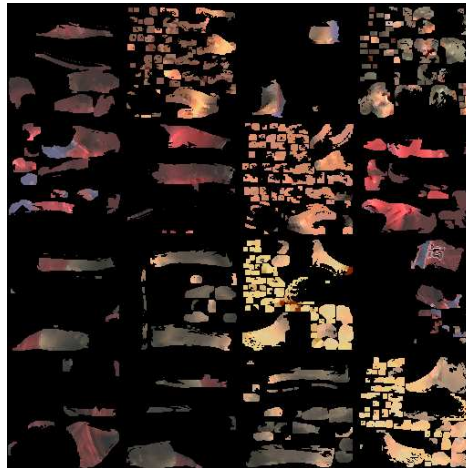
5.1.1 Hardware-Accelerated Resampling

With already accomplished hardware-accelerated projective texture mapping (Figure 5.1), we can now easily map each frame of the input streams into the



(a) Rendered with normal texture atlas.

(b) Rendered with grown texture atlas.



(c) Grown atlas after two iterations.

Figure 5.4: Growing texture atlas to remove artifacts.

texture domain using a vertex program [28]. The final vertex position is set to the 2D texture atlas coordinates of the vertex while the transformed 3D vertex position is used to compute the image position in the current frame which defines how to texture the currently rendered triangle. In order to compute visibility, we perform a traditional shadow mapping approach with the camera position used as the light source. All non-visible texels are rendered black as can be seen in Figure 5.3.

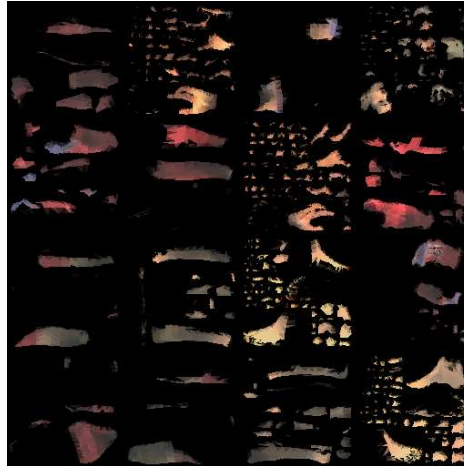


Figure 5.5: Averaged texture atlas.

5.1.2 Grown Texture Atlas

Mapping the texture atlas back onto the model results in artifacts due to OpenGL filtering (Figure 5.4(a)) that we discussed in the beginning of this chapter. Growing the texture atlas at the edges solves this problem. Some artifacts in Figure 5.4(b) are due to the low resolution of the original texture.

5.1.3 Averaged Texture Atlas

The averaged texture atlas for each time step is constructed assuming that a single triangle at time step t only consists of a single reflectance value that is averaged over all surface points in the triangle. For the light source source estimation we use the averaged texture atlases. This approximation works well for the simplification of the light source estimation process because the size of a single triangle in the texture domain is very small. Using the texture atlas parameterization, a bounding box for each triangle in the texture space is calculated. All the pixels in the triangle are found using the point in a triangle test for all the pixels in the bounding box. The point in a triangle test is implemented by calculating the barycentric coordinates for the point. A value between 0 and 1 for the barycentric coordinates shows that the point lies in the triangle. All pixels in the triangle are averaged. This averaged value is assigned to all the pixels in the triangle to

construct the averaged texture atlases (Figure 5.5).

5.2 Discussion

We described the process of texture-to-geometry registration and the parameterization of the mesh that yields to a texture atlas. Using the camera and motion parameters, the input frames are resampled into a separate texture atlas for each time step and each camera. The texture atlas parameterization remains constant for all time steps, assisting in accessing the information of each texel for the whole sequence to be used for assembling reflectance samples as discussed in Chapter 6.

Chapter 6

BRDF Reconstruction

In the previous chapter we explained the resampling process from input frame into texture space so that texture atlases for all time steps and cameras are made available. In the Free-Viewpoint Video for Human Actors [2] project, a time dependent texture map is created to render the object from novel viewpoints. In order to render the object under novel lighting conditions, we need to estimate surface reflectance properties, for which it is necessary to know the illumination conditions. We present an approach that estimates both variables simultaneously.

6.1 Light Source Estimation

Estimating the BRDF requires the lighting conditions to be known, because the reflectance model is directly dependent on the incident illumination. In Section 3.2.2 we discussed the multi-view recording process. Our studio is lit by multiple area light sources at the ceiling of the room that form a rectangular area with respect to the center of the room. We want to approximate this illumination using a configuration of point light sources distributed in a rectangular area at the ceiling of the room. Figure 6.1 shows our initial assumption of nine point light sources. Eight light sources are distributed in a rectangular way around the center of the room, while one light source is in the center. The light sources are named with reference to the center light source according to the directions of the compass. From now on we will refer to each light source by its initials, e.g. NE for

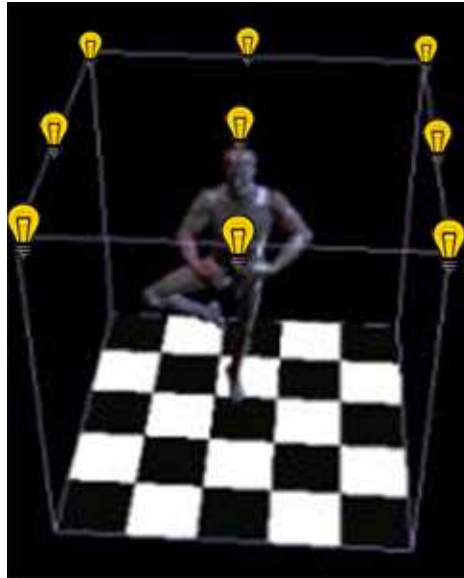


Figure 6.1: Light source distribution.

North East. The *Center* light source is referred as C . During the acquisition, the model is illuminated from all directions. We try to approximate the configuration of light sources that results in the best estimation of the BRDF. We assume it is close to the original illumination. Since we have nine light sources, approximating all possible configurations is not feasible. Some of the arbitrary configurations do not fulfill the criteria of light arriving from all directions. We only evaluate the following eight configurations:

- C
- N, E, W, S
- NE, NW, SE, SW
- NE, NW, SE, SW, C
- NE, NW, SE, SW, N, S
- NE, NW, SE, SW, E, W
- N, E, W, S, NE, NW, SE, SW

- N, E, W, S, NE, NW, SE, SW, C

The list shows that aside from the first configuration which was selected for testing purpose all other configurations are distributed evenly around the center of the room. Pseudocode for approximating the best light source configuration is given in Figure 6.2:

```

FOREACH light source configuration  $i$ 
  Assemble data with respect to  $i$ 
  Estimate the BRDF for the complete mesh
  Estimate the diffuse component  $k_d$  for each triangle
   $E_a \leftarrow$  average error for the estimated BRDF
   $E_d \leftarrow$  average error for the estimated diffuse component  $k_d$ 
   $Best \leftarrow i$  if  $E_a$  and  $E_d$  are minimum

```

Figure 6.2: Pseudocode for the approximation of the best light source configuration.

For this approximation we work on per-triangle data using the averaged texture atlases. The algorithm is explained in detail in the subsequent sections.

6.1.1 Assembling Reflectance Values

We resample the averaged texture atlases using texture atlas parameterization (Section 5.1.3). The averaged texture atlases allow us to use one reflectance sample per triangle for each time step and each camera. For a triangle we use its center in texture coordinates to obtain a reflectance value from an averaged texture atlas. The center of the triangle is chosen to avoid the boundary pixels, assuring that the sample belongs to the correct triangle. The texture parameterization does not change over time, but triangle positions (in world coordinates) may change over time. For each time step we calculate the normal \vec{n} of each triangle and its center \vec{c} . For every triangle, given known viewing conditions, i.e. eight fixed camera positions for the whole sequence, the view vector \vec{v}_n for the camera n can be found for each time step as:

$$\vec{v}_n = \vec{V}_n - \vec{c} \quad n = 1 \dots \text{number of cameras} \quad (6.1)$$

where \vec{V}_n is the position of camera n . Similarly for each time step, given the light source position \vec{L}_k , the light vector \vec{l}_k for the light source k can be found:

$$\vec{l}_k = \vec{L}_k - \vec{c} \quad k = 1 \dots \text{number of lights} \quad (6.2)$$

The reflection vector defined in Eq. (2.8) is computed for each light source k :

$$R(\vec{l}_k) = \vec{l}_k - 2(\vec{l}_k \cdot \vec{n})\vec{n} \quad (6.3)$$

For each light source, we also calculate the cosine of light, i.e. the cosine of the angle between the light vector and the normal:

$$g_k = (\vec{l}_k \cdot \vec{n}) \quad (6.4)$$

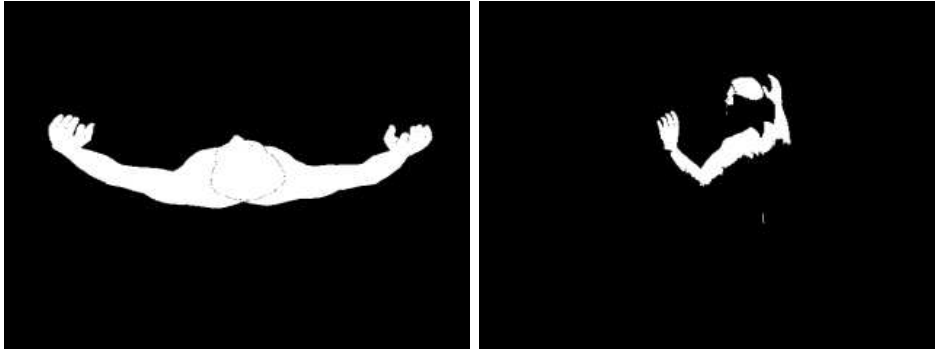
and the cosine of highlight, i.e. the cosine of the angle between the reflection vector and the view vector:

$$h_{k,n} = (R(\vec{l}_k) \cdot \vec{v}_n) \quad (6.5)$$

We define a data structure *Sample* to collect all information about a specific triangle at a specific time step for a specific camera. A *Sample*, denoted by \mathcal{S} stores a triangle's id, reflectance value, two arrays of size k , one storing cosines of light g and other cosines of highlight h . A *Sample* is constructed for each triangle for each time step and camera where it is visible. Visibility is determined by a combination of back face culling and the averaged texture atlases that are resampled considering visibility using the geometry model. For reliable light source visibility estimation, texture atlases considering visibility from light sources are generated for each time step. They are used to determine the number of light sources contributing to the current reflectance sample. Figure 6.3 shows a shadow atlas generated for the *Center* light source.

6.1.2 BRDF Estimation

In Section 2.7.3 we discussed empirical BRDF models and presented the extended Phong model (Eq. (2.9)). We fit the parameters of this model to our data. The



(a) Light source view.

(b) Rendered from different viewpoint.



(c) Texture atlas considering visibility.

Figure 6.3: Resampling the shadow atlas from the center light source.

Phong model is non-linear in its parameters, therefore a non-linear optimization method has to be used to best fit the parameters to our data. We use the Levenberg-Marquardt optimization [16] to estimate the parameters of the Phong model. In Section 6.2.3 we show that an analysis by synthesis approach using Powell's method [16] yields similar results.

Average BRDF

We first fit an average BRDF to all our measured data. We fit a separate BRDF to each color channel. We need to estimate 9 parameters, 3 (k_d , k_s , k_e) for each color channel.

We initialize the fitting process with parameters that correspond to an average BRDF. The iterative fitting process stops if the error measure goes below a certain threshold or if it reaches the maximum number of iterations.

Diffuse BRDF

In order to estimate the diffuse component k_d for each triangle, the specular component obtained from the average BRDF is subtracted from all *Samples*:

$$reflectance - specular = k_d * \sum_i^k (\vec{l}_i \cdot \vec{n}) \quad (6.6)$$

Eq. (6.6) presents the generic equation for the diffuse component k_d . The triangle id in each \mathcal{S} is used to couple all its *Samples*. The number of equations yield an over-determined system of linear equations for all the frames where the triangle is visible. The diffuse component (k_d) is obtained as a least square solution of the system of linear equations.

6.1.3 Error Estimation

The reliability of the estimated average and individual diffuse component k_d can be found by computing the sum of the squared difference between the measured reflectance and the evaluated reflectance. The sum is normalized with the total number of reflectance samples. The square of the difference is computed for all color channels and then averaged to get the mean error. Given a measured reflectance sample r and an evaluated BRDF f , the error for the average BRDF can be found:

$$E_a = \frac{1}{|\mathcal{S}|} \sum_i^{|\mathcal{S}|} (r_i - f_i)^2 \quad i = 1 \dots \text{number of triangles} \quad (6.7)$$

Where $|\mathcal{S}|$ is the total number of *Samples*. For the diffuse error, the sum of the squared difference is computed for all *Samples* of every triangle using its diffuse component. It is normalized by the number of *Samples* in a specific triangle and the total number of triangles T :

$$E_d = \frac{1}{T} \sum_j \frac{1}{|\mathcal{S}_T|} \sum_i (r_{j,i} - f_{j,i})^2 \quad \begin{cases} i = 1 \dots \text{number of samples in triangle T} \\ j = 1 \dots \text{number of triangles} \end{cases} \quad (6.8)$$

Visualizations of the estimated BRDFs and their respective errors are shown in Figures 6.4 and 6.5. The BRDFs are estimated using the third light source configuration (NE, NW, SE, SW). The error is color encoded. Dark blue represents low error and dark red represents high error, while other colors lie in between. As expected error drops for the individual diffuse component k_d .



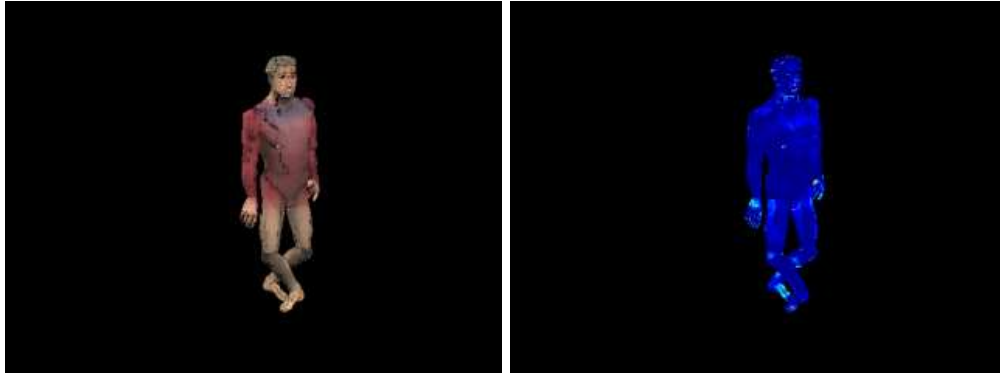
(a) Average BRDF.

(b) Average general error.

Figure 6.4: Average BRDF estimation.

6.1.4 Best Light Source Configuration

The computed errors shows the relationship of a light source configuration with the original illumination conditions. Smaller error represents higher correspondence. Table 6.1 shows the nine light source configurations, the corresponding number of lit triangles, the average error E_a and the diffuse error E_d . Errors are



(a) Diffuse BRDF.

(b) Diffuse error.

Figure 6.5: Diffuse BRDF estimation.

averaged for the whole sequence and range from 0 to 1.

| Configuration | Triangles | Average Error | Diffuse Error |
|-------------------------------|-----------|---------------|---------------|
| C | 21274 | 0.259569 | 0.1032580 |
| N, E, W, S | 26140 | 0.209228 | 0.0875627 |
| NE, NW, SE, SW | 26545 | 0.172890 | 0.0706268 |
| NE, NW, SE, SW, C | 26545 | 0.207112 | 0.0856741 |
| NE, NW, SE, SW, N, S | 26546 | 0.186107 | 0.0782518 |
| NE, NW, SE, SW, E, W | 26545 | 0.194521 | 0.0850380 |
| N, E, W, S, NE, NW, SE, SW | 26546 | 0.196265 | 0.0838712 |
| N, E, W, S, NE, NW, SE, SW, C | 26546 | 0.212322 | 0.0918319 |

Table 6.1: Average and diffuse errors for light source configurations.

According to our algorithm (Figure 6.2), the third light source configuration (NE, NW, SE, SW) is selected as the best configuration that will be further used to assemble per-textel data and estimate surface reflectance properties. It can be seen in Table 6.1 that the number of lit triangles does not change with an increasing number of light sources, whereas the error changes.

6.2 Reflectance Measurement

In the previous section we approximated the illumination conditions in our studio. Solving for illumination was dependent on the BRDF estimation (Section 6.1.2). For the realistic BRDF estimation we should consider data from all surface points. Per-triangle BRDF estimation ignores the per pixel variation (Figure 6.5(a)). We only have one average specular component for all reflectance samples. Real world objects are composed of different materials with unique reflectance properties. For truly realistic representation we should obtain a BRDF for each texel, but given the sparse valid samples for each texel, it is not possible to fit a reliable BRDF to the input data. This is due to the lack of enough samples with specular part for each texel because reflection happens only in a small solid angle which might not be observed sufficiently. In order to approximate a BRDF for each texel, we classify all texels into different clusters and approximate a BRDF for each cluster. Afterwards the specular part of the estimated cluster BRDF is subtracted from the samples of the respective cluster and the diffuse component is re-estimated for each texel.

6.2.1 Clustering

We use k -means clustering [10][11] to classify each texel into a specific cluster. In this procedure, we start with the assumption that our object is composed of three clusters, i.e. clothing, skin, and hair. We pick three reflectance samples from each cluster:

- Clothing = RGB(144, 71, 72)
- Skin = RGB(160, 135, 109)
- Hair = RGB(76, 74, 67)

These reflectance samples are defined as the base reflectance samples for each cluster. The k -means clustering algorithm is applied on the averaged samples of each texel. The sum of the squared distance of each averaged sample from the base reflectance samples is computed. Each texel is classified according to the minimum distance. The mean value of each newly formed cluster is computed

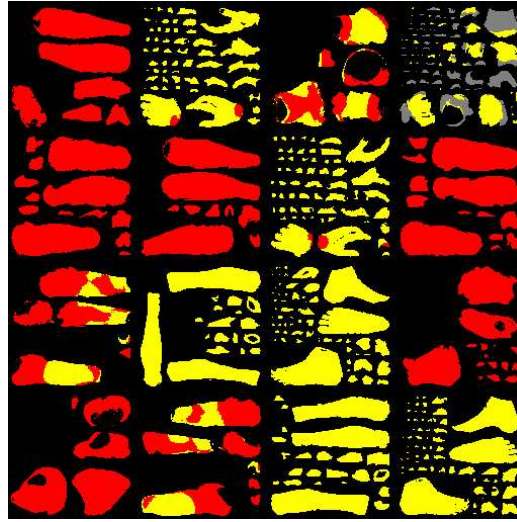


Figure 6.6: Clustering texels into three materials, red, yellow, and grey represent clothing, skin, and hair respectively.

and defined as the base reflectance sample of that cluster. The algorithm is repeated unless the change in the number of samples in each cluster is very small. This method is semi-automatic because we input three reflectance samples, representing each cluster, and the number of materials in each body segment, e.g. only two materials skin and hair are present in the skull segment. Figure 6.6 shows the clustered texels, red, yellow and grey denoting clothing, skin and hair respectively.

6.2.2 Assembling Samples

We improve the concept of the *Sample* data structure presented in Section 6.1.1. Since we need to assemble *Samples* for each texel in the texture atlas, we use the original resampled texture atlases instead of the averaged ones. In the *Sample* data structure, the triangle id is discarded and three new fields are added: position \vec{x} to store the texel position in texture coordinates, time step t , and cluster c . Cluster c is used in per-cluster BRDF estimation, position \vec{x} is used in per-texel diffuse estimation and time step t is used in constructing a diffuse correction map for each time step (Section 7.1.2). Following the same procedure presented in Section 6.1.1, a *Sample* is constructed for each texel per time step and camera

where it is visible. Visibility is determined by back face culling and texture atlases, while the shadow atlases (Figure 6.3) are used to determine the light source contribution.

6.2.3 BRDF Fitting

We follow the same approach presented in Section 6.1.2 to fit the extended Phong model parameters (Eq. (2.9)) to our data.

Cluster BRDF

Two different optimization methods are used to estimate the parameters of the Phong model on our input data, i.e. Levenberg-Marquardt optimization [16] and Powell’s method [16]. In an iterative analysis-by-synthesis approach, using estimated parameters by Powell’s method we simulate reflectance samples to minimize the energy function. Powell’s method is initialized with parameters that correspond to an average BRDF. The estimated parameters from the method are used to evaluate reflectance samples. The energy function returns the sum of the squared difference between the measured reflectance and the evaluated reflectance. The iterative fitting process stops if the error measure goes below certain threshold or if it reaches maximum number of iterations. Table 6.2 shows the estimated BRDF for the clothing material. The results from Levenberg-Marquardt optimization and Powell’s method are very close, suggesting that either method can be used for similar estimations.

| Optimization Method | k_d | k_s | k_e |
|---------------------|----------|-----------|---------|
| Levenberg Marquardt | 0.546393 | 0.0124035 | 3.12680 |
| Powell | 0.544803 | 0.0164708 | 3.06849 |

Table 6.2: Comparison of the two optimization methods.

Diffuse BRDF

The estimated specular part of each cluster is subtracted from the reflectance value of all *Samples* belonging to the same cluster (Eq. (6.6)). All the *Samples* of a single texel are used to obtain a least square solution for its diffuse component (Section 6.1.2).

6.3 Discussion

We focused on approximating illumination conditions and reflectance measurement. Using a method that involves the BRDF estimation, we approximated the best light source configuration. Considering that video streams were not recorded in a room lit by point light sources, our approximation introduces some errors.

We use two different methods for the BRDF estimation. The estimated parameters from Levenberg Marquardt and Powell's method are very close. For Powell's method we minimize the sum of the square difference between the rendered and observed reflectance samples. This is done in the energy function. Levenberg Marquardt method internally incorporates it by calculating the chi-square merit function. The results suggest that either method can be used for the estimation process.

The best light source configuration is used for per-texel BRDF estimation. All surface points are clustered into three categories. For reliable clustering we use a semi-automatic procedure. The specular component is approximated for each cluster. We estimate a diffuse component for each texel. This allows us to represent the estimated BRDFs as texture atlases and facilitates in the rendering process. In the next chapter we will explain the rendering process and present the results.

Chapter 7

Results

In the previous chapter we explained the process of BRDF estimation. For every surface point, we know its estimated BRDF and 2D texture coordinates. This information is used in the rendering process to construct the BRDF atlases.

7.1 Rendering

Rendering of the model with a grown texture atlas is explained in Section 5.1.2. A similar approach is used to render the model with the estimated per-texel BRDFs.

7.1.1 Assembling the BRDF atlases

We construct a texture atlas for each of the BRDF parameters (k_d , k_s , k_e). The specular and exponent components (k_s , k_e) are taken from per-cluster BRDFs, whereas the diffuse component (k_d) is available per-texel (Figure 7.1). These floating point atlases are grown as explained in Section 5.1.2 to prevent rendering artifacts. The whole sequence is now described by the motion parameters of the body model and three texture atlases. The texture information has been reduced from per-frame texture atlases to three atlases that are valid for the whole sequence. Figure 7.2 shows the model rendered using the BRDF atlases. Per pixel variation due to the individual diffuse component for each texel can be observed. The model appearance does not look realistic due to the limitations of the Phong

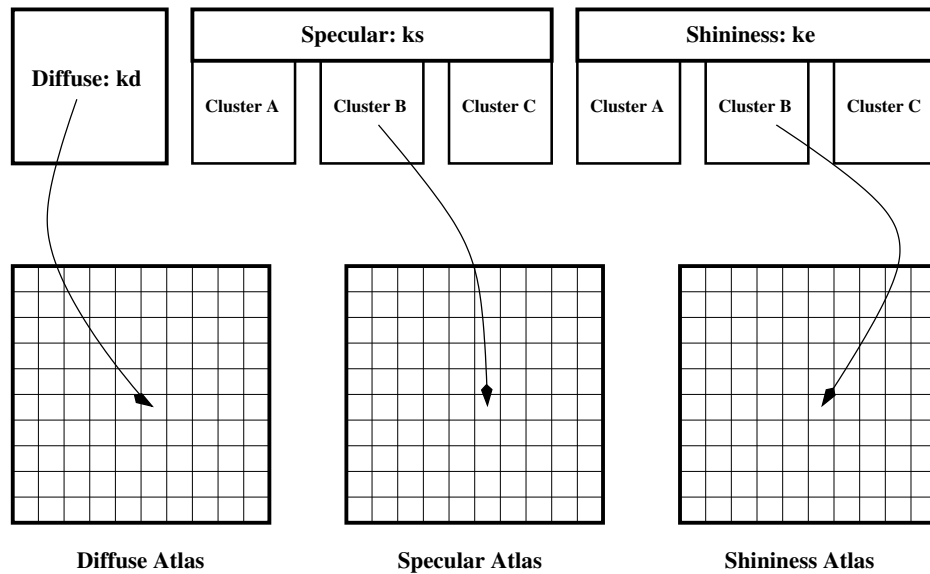


Figure 7.1: Assembling the BRDF Atlases.



Figure 7.2: The model rendered with the BRDF atlases.

model and the low quality of the model. The model does not accurately represents the human actor. Since it is naked whereas the actor wears clothes, resulting in normals point in the wrong directions for some regions. Some artifacts that can be observed are due to mesh inaccuracies and the low resolution of the input data. Especially, data at depth discontinuities and in concave regions is a source of error because we do not consider interreflections. The rendering is hardware accelerated, the BRDF atlases are loaded as textures, per-pixel normals are calculated, and the BRDF model is evaluated in a fragment shader for each pixel separately.

7.1.2 Correction atlas

The BRDF atlases allows the model to be rendered under novel illumination conditions and from novel viewpoints. The quality of the reconstructed model is lower due to the the low quality of the input data and mesh inaccuracies. The reconstructed BRDFs can be used to render the model accurately under the original light source configurations but from novel viewpoints by constructing a diffuse correction atlas for each time step. We assume that the specular component is correct for the whole sequence, whereas for each time step we can solve for the diffuse component in the input data. The difference between this diffuse data and the diffuse atlas is stored in a correction atlas for each time step. This correction atlas is loaded as a texture and used to correct the diffuse atlas dynamically in the fragment shader. Figure 7.3 shows pseudocode for generating the correction atlas:

```

FOREACH time step  $t$ 
  RdAtlas ← RENDER using diffuse atlas ( $k_d \sum_i^k (\vec{l}_i \cdot \vec{n})$ )
  IlluminationAtlas ← RdAtlas / kdAtlas
  OdAtlases ← Original texture atlases at  $t$  – Specular contribution
  AdAtlas ← Average all OdAtlases
  OkdAtlas ← AdAtlas / IlluminationAtlas
  Correction Atlas ← kdAtlas – OkdAtlas

```

Figure 7.3: Pseudocode for generating the correction atlas.

Figure 7.2 shows the model rendered using the BRDF atlases. The rendered reflectance (f_r) is computed from the estimated BRDF parameters:

$$f_r = k_d \sum_i^k (\vec{l}_i \cdot \vec{n}) + k_s \sum_i^k (R(\vec{l}_i) \cdot \vec{v})^{k_e} \quad (7.1)$$

The observed reflectance (f_o) is computed from the original texture atlases:

$$f_o = k_d^o \sum_i^k (\vec{l}_i \cdot \vec{n}) + k_s^o \sum_i^k (R(\vec{l}_i) \cdot \vec{v})^{k_e^o} \quad (7.2)$$

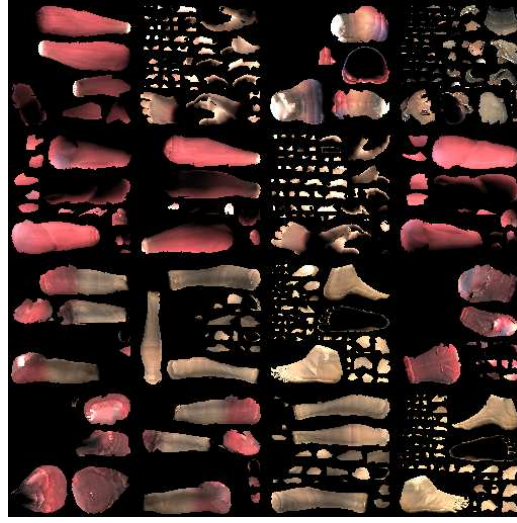


Figure 7.4: The rendered diffuse atlas.

The error (ϵ) is the difference between the rendered and the observed reflectance:

$$\epsilon = f_r - f_o \quad (7.3)$$

Since we want to correct the diffuse component (k_d), k_s and k_e are assumed correct ($k_s = k_s^o$, $k_e = k_e^o$):

$$\epsilon = f_r - f_o = (k_d - k_d^o) \sum_i^k (\vec{l}_i \cdot \vec{n}) \quad (7.4)$$

In the first step of the algorithm we render the model with the k_d atlas only and resample it into a diffuse texture atlas as shown in Figure 7.4. The illumination atlas is computed to store the light source contribution ($\sum_i^k (\vec{l}_i \cdot \vec{n})$). To compute the observed diffuse reflectance, the estimated specular component is subtracted from the original texture atlases. Each texel in the correction atlas (c_r) stores the difference between k_d and k_d^o :

$$c_r = \frac{\epsilon}{\sum_i^k (\vec{l}_i \cdot \vec{n})} \quad (7.5)$$

This difference is added to k_d dynamically at each time step for the accurate rep-



(a) Input Frame.

(b) The mode rendered using the reconstructed BRDF and the correction atlas.

Figure 7.5: Comparison between the original and the reconstructed image.

resentation of the diffuse component (Figure 7.5). Table 7.1 presents the average error for each camera after the model has been rendered to the original views with the BRDF and the correction atlases. The error is computed by normalizing the square of the difference between reflectance values in the original texture atlases and the corresponding reflectance values in the resampled atlases of the model rendered with the BRDF and the correction atlases. The error reduces using the correction atlases. It is not exactly zero because we averaged the observed diffuse atlases which introduced a small error.

| Camera | BRDF Atlas | Correction Atlas |
|--------|------------|------------------|
| 1 | 0.0315510 | 0.00559267 |
| 2 | 0.0318326 | 0.00832741 |
| 3 | 0.0359250 | 0.00864765 |
| 4 | 0.0347399 | 0.00756801 |
| 5 | 0.0347740 | 0.00901620 |
| 6 | 0.0365766 | 0.00935371 |
| 7 | 0.0316859 | 0.00618237 |
| 8 | 0.0330833 | 0.00843030 |

Table 7.1: Average error for the BRDF and the correction atlas rendering.



Figure 7.6: The model rendered using the BRDF atlases.

7.2 Discussion

The reconstructed BRDFs allow us to render the model from novel views and under novel illumination conditions with compact representation. We present the rendering process from the reconstructed BRDFs. We also construct the correction atlases for accurate rendering under the original light source configuration. Figure 7.6 shows the model rendered using the BRDF atlases from a novel viewpoint. Figure 7.7 shows a sequence rendered using the BRDF atlases. A single light source moves around the model while the camera changes after every 7th frame so that the effect of the moving light source can be observed. Figure 7.8 shows the model rendered using the BRDF and the correction atlases from novel viewpoints.

The model rendered with the BRDF atlases does not appear realistic. The approximation of area lights with point light sources introduces some errors. The limitations of the Phong model result in slightly different appearance of materials. The low quality of the geometric model plays a significant part for the inaccurate appearance. The clothes wore by the human actor are not modeled resulting in normals pointing in wrong directions. The correction atlases significantly reduce the error. They also introduce false shadows in certain regions. Some errors still remain and visual artifacts are observable, mainly due to the low quality of the input data and the triangle mesh.

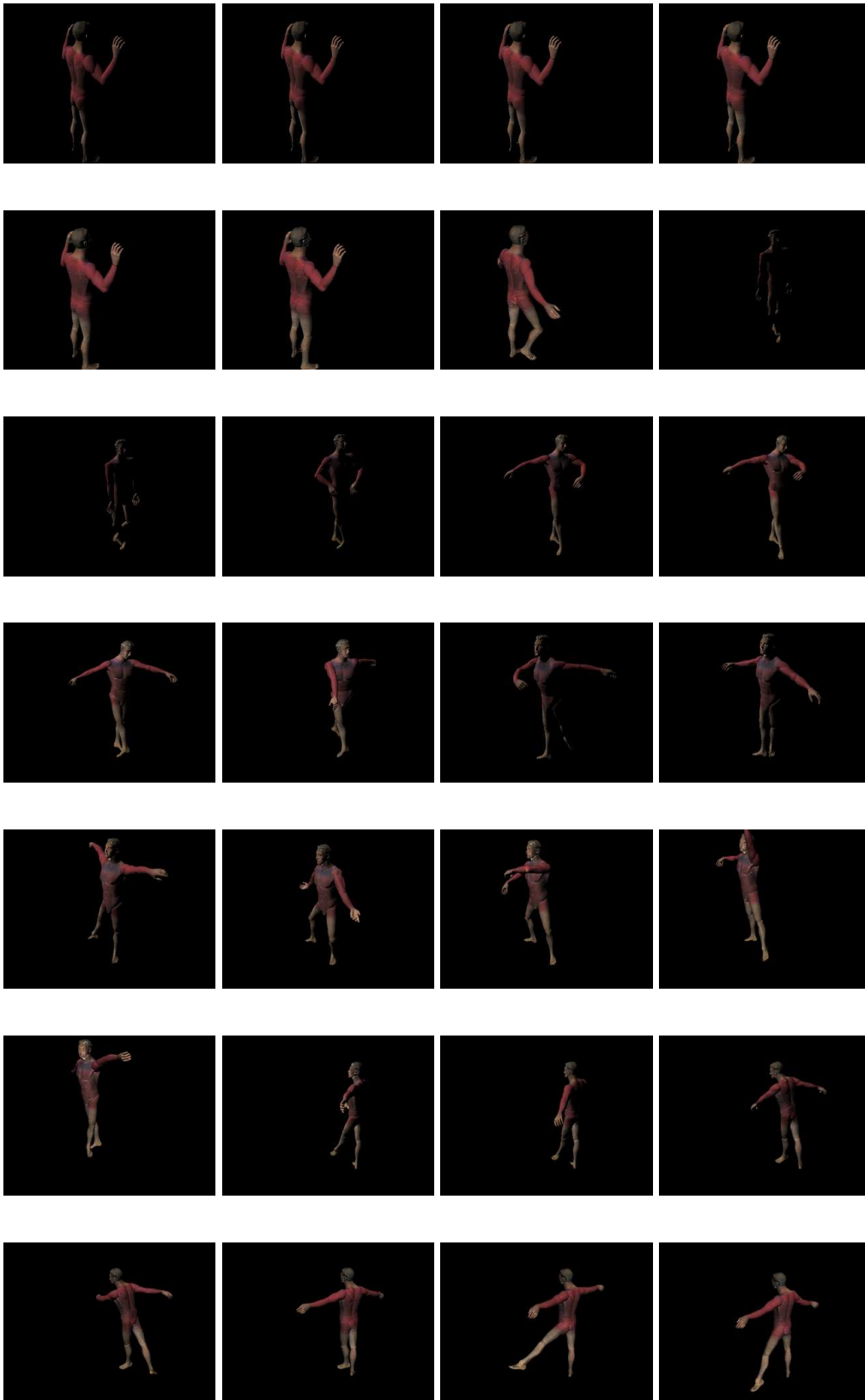


Figure 7.7: The model rendered under changing illumination conditions. A single light source moves around the model. The camera changes after every 7th frame.

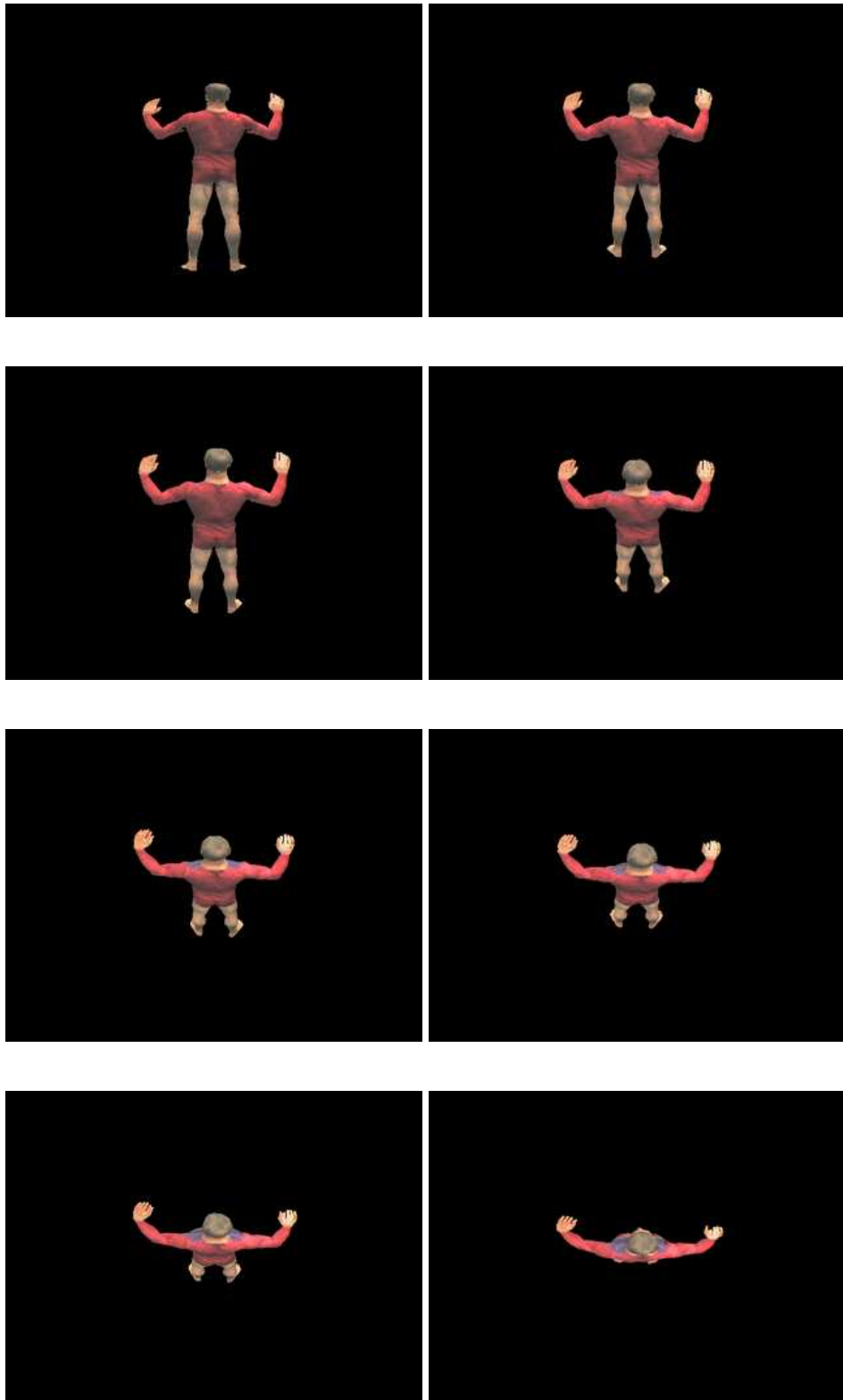


Figure 7.8: The model rendered from novel viewpoints using the correction atlases.

Chapter 8

Conclusions & Future Work

Inverse rendering has received a lot of attention in the last few years. Traditionally, very high quality images of a static object are used to measure surface reflectance properties. An obvious extension is to work on dynamic objects. This requires algorithms to determine reliable visibility and texture-to-image registration for the dynamic geometry. Although it is possible to acquire very high quality static images using digital cameras, current video cameras are still very limited in this domain. Limited frame rate further degrades the quality of the acquired video data.

We have presented a system for reconstructing surface reflectance properties of a dynamic object from multi-video recordings. Our work encompasses two problems in the domain of inverse rendering, i.e. inverse lighting and inverse reflectance. We presented an algorithm to approximate the best light source configuration for our studio and used this configuration to estimate per-texel BRDFs. We demonstrated that despite the low quality of the input data, reasonably realistic reconstruction can be achieved.

We used data from the Free-Viewpoint Video of Human Actors [2] project that provides video streams, a geometric model, and a framework for texture-to-geometry registration. Our algorithm relies on all input data mapped into the texture domain by defining a texture atlas that maps the 3D geometry into the 2D texture domain. A texture atlas for each body part is constructed and then stitched together to form a single texture. Since the texture parameterization is

consistent over time it allows us to access reflectance samples for each surface point independent of its geometrical transformations over time.

We approximated the area light sources in our studio with different configurations of point light sources and selected the configuration that resulted in the minimum error for the estimated per-triangle BRDF. For each light source configuration, data is assembled considering visibility from each viewpoint and from the light sources. An average BRDF for all samples and a diffuse component for each triangle are estimated. Errors for the average BRDF and the diffuse components are evaluated for each light source configuration. The configuration with the minimum error is chosen for further per-textel BRDF estimation.

All surface points are clustered into three categories based on their average reflectance value using k -means clustering. A single BRDF is estimated for each cluster. For each texel a separate diffuse component is estimated. For each BRDF parameter a texture atlas is constructed, storing corresponding estimated parameters for each surface point. A correction atlas is constructed for each time step that stores the difference between the diffuse component of the estimated BRDF and the diffuse component computed from the input data. Hardware-accelerated rendering is used to render the model with the BRDF atlases. The correction atlases are used to dynamically correct the diffuse BRDF parameter in each time step.

During this thesis we explored different problems. While we presented one solution, there are other directions that could provide different solutions and should be investigated. Working with video streams has its advantages and disadvantages. On one hand we have abundant data, whereas due to the low quality of the video streams and a small frame rate, temporal aliasing seriously degrades the reliability of the obtained reflectance samples. One obvious alternative is to work with high quality data, but in any case it is worth exploring some reliability criteria for reflectance samples over time that should only accept potentially correct samples for a surface point.

The quality of the mesh should be considerably higher to increase the accuracy of the surface parameterization and the resampling. A geometric model that closely represents the object can increase the reliability of reflectance samples. The Phong model has some disadvantages especially the appearance of the rendered model is not very realistic. A physics- based BRDF model should result in

the better estimation and appearance.

Programmable graphics hardware opens new ways to solve problems. We exploited graphics hardware for the tasks that would otherwise be very expensive to perform in software. Still current generation graphics hardware is limited in terms of flexibility. The arrival of Pixel and Vertex shader 3.0 models with the next generation hardware will expand the horizon. Not only could it be used to evaluate more complex BRDF models, but it may also be possible to develop new algorithms specific to the graphics hardware that can lead to more reliable and much faster reflectance measurement.

Bibliography

- [1] Ronen Basri and David Jacobs. Photometric stereo with general, unknown lighting. In *Proc. CVPR*, pages 374–381, December 2001.
- [2] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. In *Proc. SIGGRAPH*, pages 569–577, July 2003.
- [3] W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel. Illuminating Micro Geometry Based on Precomputed Visibility. In *Proc. SIGGRAPH*, pages 455–464, July 2000.
- [4] James T. Kajiya. The rendering equation. In *Proc. SIGGRAPH*, pages 143–150, August 1986.
- [5] John K. Kawai, James S. Painter, and Michael F. Cohen. Radioptimization - goal based rendering. In *Proc. SIGGRAPH*, pages 147–154, August 1993.
- [6] J. Lambert. *Photometria Sive de Mensura et Gradibus Luminus, Colorum et Umbrae*. Eberhard Klett, 1760.
- [7] Hendrik P. A. Lensch. *Efficient, Image-Based Appearance Acquisition of Real-World Objects*. PhD thesis, Max-Planck-Institut für Informatik, 2004.
- [8] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.*, 22(2):234–257, 2003.
- [9] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, August 1996.
- [10] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, IT-28:129–137, 1982.

- [11] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, volume 1, 1967.
- [12] Sebastian Magda, David Kriegman, Todd Zickler, and Peter Belhumeur. Beyond lambert: Reconstructing surfaces with arbitrary BRDFs. In *Proc. ICCV*, pages 391–399, July 2001.
- [13] Stephen R. Marschner. *Inverse Rendering in Computer Graphics*. PhD thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, 1998.
- [14] Harry Moseley and David H Sliney. Radiometric quantities and units. *Physics in Medicine and Biology*, 42(5):762, 1997.
- [15] B.-T. Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18(6):311–317, June 1975.
- [16] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge Univ. Press, 2nd ed. edition, 1994.
- [17] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proc. SIGGRAPH*, pages 117–128, August 2001.
- [18] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *Proc. SIGGRAPH*, pages 409–416, August 2001.
- [19] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, and Paul Haeberli. Fast shadow and lighting effects using texture mapping. In *Proc. SIGGRAPH*, pages 249–252, July 1992.
- [20] W. M. Silver. Determining shape and reflectance using multiple images. Master’s thesis, MIT, Cambridge, MA, 1980.
- [21] Olga Sorkine, Daniel Cohen-Or, Rony Goldenthal, and Dani Lischinski. Bounded-distortion Piecewise Mesh Parameterization. In *IEEE Visualization*, pages 355–362, 2002.
- [22] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9):1105–1114, 1967.
- [23] Roger Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. CVPR*, pages 364–374, June 1986.

- [24] G. Ward Larson. Measuring and modeling anisotropic reflection. In *Proc. SIGGRAPH*, pages 265–272, July 1992.
- [25] Wikipedia.org. Light. <http://en.wikipedia.org/wiki/Light>.
- [26] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface light fields for 3D photography. In *Proc. SIGGRAPH*, pages 287–296, July 2000.
- [27] R. Woodham. Analysing Images of Curved Surfaces. *Artificial Intelligence*, 17:117–140, 1981.
- [28] Gernot Ziegler, Hendrik P. A. Lensch, Naveed Ahmed, Marcus Magnor, and Hans-Peter Seidel. Multi-video compression in texture space. In *Proc. ICIP*, October 2004, to appear.