INVERSE RENDERING FOR COMPUTER GRAPHICS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Stephen Robert Marschner

August 1998

INVERSE RENDERING FOR COMPUTER GRAPHICS

Stephen Robert Marschner, Ph.D.
Cornell University 1998

Creating realistic images has been a major focus in the study of computer graphics for much of its history. This effort has led to mathematical models and algorithms that can compute *predictive*, or *physically realistic*, images from known camera positions and scene descriptions that include the geometry of objects, the reflectance of surfaces, and the lighting used to illuminate the scene. These images accurately describe the physical quantities that would be measured from a real scene. Because these algorithms can predict real images, they can also be used in inverse problems to work backward from photographs to attributes of the scene.

Work on three such *inverse rendering* problems is described. The first, *inverse lighting*, assumes knowledge of geometry, reflectance, and the recorded photograph and solves for the lighting in the scene. A technique using a linear least-squares system is proposed and demonstrated. Also demonstrated is an application of inverse lighting, called re-lighting, which modifies lighting in photographs.

The second two inverse rendering problems solve for unknown reflectance, given images with known geometry, lighting, and camera positions. *Photographic texture measurement* concentrates on capturing the spatial variation in an object's reflectance. The resulting system begins with scanned 3D models of real objects and uses photographs to construct accurate, high-resolution textures suitable for physically realistic rendering. The system is demonstrated on two complex natural objects with detailed surface textures.

Image-based BRDF measurement takes the opposite approach to reflectance measurement, capturing the directional characteristics of a surface's reflectance by measuring the *bidirectional reflectance distribution function*, or BRDF. Using photographs of an object with spatially uniform reflectance, the BRDFs of paints and papers are measured with completeness and accuracy that rival that of measurements obtained using specialized devices. The image-based approach and novel light source positioning technique require only

general-purpose equipment, so the cost of the apparatus is low compared to conventional approaches. In addition, very densely sampled data can be measured very quickly, when the wavelength spectrum of the BRDF does not need to be measured in detail.

Biographical Sketch

The author was born on November 29, 1971. From 1989 to 1993 he studied at Brown University in Providence, Rhode Island, where he received an Sc. B. degree in Mathematics/Computer Science. He then moved to Ithaca, New York, to study at Cornell University from 1993 to 1998. He received his doctorate from Cornell in 1998.

for Heidi

Acknowledgements

My first thanks go to my advisor, Don Greenberg, for the support and encouragement he gave me during my stay at Cornell. His constant confidence in me and my work has been an inspiration, and he has surrounded me with the people and technology that made that work possible. I would also like to thank Len Gross and Charlie Van Loan, whom I enjoyed working with as my minor advisors. Charlie, in his fun and inspiring classes, taught me most of what I know about numerical computation, much of which has come into use in this dissertation. He also pointed me toward the right acceleration scheme for the inverse lighting problem in Chapter 3. I am greatly indebted to Ken Torrance, whose wisdom and encouragement are responsible for convincing me to pursue the idea of image-based BRDF measurement.

During my time in the Program of Computer Graphics I had the pleasure of working with many wonderful people. I thank Richard Lobb and Pete Shirley for being my informal advisors during different stages of my research. Pete also taught me a great deal about rendering, and the code from the *eon* renderer he created is responsible, one way or another, for every rendered image in this dissertation. Without the many valuable discussions and suggestions from Bruce Walter, Eric Lafortune, and Steve Westin, this dissertation would not be what it is. Steve also helped extensively with my BRDF measurements, and Eric drove the gonioreflectometer for me. Thanks to Jim Ferwerda for many enjoyable hours working together on the video lab. Thanks to Hurf Sheldon and Mitch Collinsworth for keeping the machines up and connected, and to Jonathan Corson-Rikert, Ellen French, Linda Stephenson, and Peggy Anderson for five years of cheerful administrative support. Thanks, Jonathan, for finding the rock for Chapter 4 and lugging it all the way back from Maine. Many thanks to Bruce, Ben Trumbore, Don, and especially Eric for their generosity and logistical support during my last month in Ithaca. Tanks also to James Durkin for discussions and help with the software and Dan Kartch for the LaTeX installation.

I would like to thank my parents for buying me that Apple II+ way back when and for believing in me.

I owe the deepest gratitude of all to Heidi, my wife and best friend, who has cheerfully accommodated all manner of long hours and inconvenient absences. Her unconditional love and constant encouragement are what keep me going.

This work was supported by the NSF Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219) and by NSF Grant ACI-9523483. I gratefully acknowledge the support of the Hewlett Packard Corporation, on whose workstations this work was carried out. I would also like to thank John Meyer and all my colleagues at Hewlett Packard Laboratories for generously accommodating my work on this dissertation during my research internship there.

Table of Contents

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vii
List of Tables . . . x
List of Figures . . . xi

1 Introduction **1**

2 Background **3**
2.1 Mathematical Preliminaries . . . 3
2.2 Radiometry . . . 5
2.3 The Bidirectional Reflectance Distribution Function . . . 8
2.3.1 Definition and basic properties . . . 8
2.3.2 Common reflectance phenomena . . . 10
2.4 Rendering . . . 13

3 Inverse Lighting **17**
3.1 Problem Statement . . . 18
3.2 Prior Work . . . 18
3.3 Basic Least-squares Solution . . . 19
3.4 Regularized Solution . . . 21
3.4.1 Accelerating using the GSVD . . . 24
3.5 Re-lighting . . . 25
3.6 A Test with a Synthetic Photograph . . . 26
3.7 A Test with a Rigid Object . . . 30
3.7.1 Camera Calibration . . . 30
3.7.2 Results . . . 31
3.8 Tests on Human Faces . . . 31
3.8.1 Filtering out unreliable data . . . 31
3.8.2 2D image warps to correct registration . . . 33
3.8.3 Results . . . 33
3.9 Conclusion . . . 35
3.9.1 Future work . . . 39

4 Photographic Texture Measurement 41
4.1 Prior Work 42
4.1.1 Texture mapping 42
4.1.2 3D scanning and reflectance modeling 43
4.2 Texture Representation 45
4.3 Estimating Reflectance 49
4.3.1 Estimating with a Lambertian BRDF model 51
4.3.2 Using non-Lambertian models 53
4.4 A Synthetic Example 54
4.5 Measurement Setup 54
4.6 Results 58
4.7 Future Work 60
4.8 Conclusion 62

5 Image-based BRDF Measurement 73
5.1 Overview of Method 73
5.1.1 Sampling patterns 76
5.2 Prior Work 78
5.3 Apparatus 80
5.3.1 The primary camera 81
5.3.2 The test samples 81
5.3.3 Calibration 82
5.4 Data Processing 83
5.5 Results 84
5.5.1 Incidence plane measurement 84
5.5.2 Full isotropic measurements 85
5.6 Mapping the BRDF Domain to 3-space 94
5.6.1 A simple cylindrical mapping 107
5.6.2 A mapping for visualization 109
5.6.3 A mapping for resampling 110
5.7 BRDF Resampling 113
5.7.1 Local polynomial regression 114
5.7.2 Reconstructing using ψ_3 116
5.8 Conclusion 116
5.8.1 Future work 117

6 Conclusion 119

A Camera Calibration 121
A.1 Geometric Calibration 121
A.2 Radiometric Calibration 124

B Bundle Adjustment 127

C Calibration Targets 129

D Cameras 135

E The Cyberware Scanner **139**

F BRDF Measurement Procedure **141**

Bibliography **143**

List of Tables

4.1 Geometric considerations affecting the reliability of estimates of diffuse reflection and the weighting factors used to account for them. 52

5.1 Summary of error measures for several accuracy tests. 86

List of Figures

2.1 Coordinate system for directions at a surface. 4
2.2 The measures σ and μ. 6
2.3 Measuring radiance for a surface element $d\mathbf{x}$ and a solid angle element $d\omega$. 6
2.4 Rotating a configuration about the surface normal. 10
2.5 The relationship between off-specular reflection and increasing specularity. 12

3.1 The data flow in the inverse lighting algorithm. 22
3.2 A 32-light basis set, drawn on the directional sphere. 23
3.3 The data flow in the re-lighting system. 27
3.4 The data flow in the re-lighting system. 28
3.5 The results of testing the re-lighting algorithm on a synthetic photograph. 29
3.6 The re-lit image from Figure 3.5 (e) compared with an image rendered directly from the new lighting. 30
3.7 Re-lighting a photograph of a rigid object. 32
3.8 Results of filtering range data. 34
3.9 Re-lighting a photograph of a human face. 35
3.10 Using re-lighting as an aid to compositing. 36
3.11 The results of re-lighting a face to move the light from the front to the side. 37
3.12 The results of re-lighting a face to move the light from the front to above. . 38

4.1 How texture patches collectively cover the object. 46
4.2 Building texture patches on a sphere. 47
4.3 The harmonic map from a texture patch to the unit circle. 48
4.4 Estimating reflectance from many views of an object. 50
4.5 Pseudocode for the texture map construction algorithm. 52
4.6 The synthetic photographs used to illustrate the texture mapping algorithm 55
4.7 The texture maps computed from the images in Figure 4.6. 56
4.8 A rendering of the sphere with the computed textures mapped onto its surface. 56
4.9 The setup used for photographing the objects. 57
4.10 A typical set of camera positions for texture measurement. 59
4.11 The geometry of the rock scan. 63
4.12 The texture patches on the rock model. 64
4.13 Some representative photographs from the set of 16 used to compute texture maps for the rock. 64
4.14 The 42 texture maps computed for the rock model. 65
4.15 A comparison between the textured rock model and the actual rock. 66
4.16 The geometry of the squash scan. 67

4.17 The texture patches on the squash model. 68
4.18 Some representative photographs from the set of 24 used to compute texture maps for the squash. 68
4.19 The nine texture maps computed for the squash model. 69
4.20 A comparison between the textured diffuse squash model and the actual object. 70
4.21 The squash model with the fitted spatially uniform specular component. . 71
4.22 The matrix structure of the linear subproblem at the core of the proposed regularized reflectance estimation system. 71

5.1 Device A for measuring BRDFs. 74
5.2 Device B for measuring BRDFs. 75
5.3 An image-based BRDF measurement device. 77
5.4 Measuring incidence-plane reflection from a cylindrical sample. 77
5.5 The angle between the source and camera directions remains nearly constant for each image. 86
5.6 The (θ_i, θ_e) locations of the sheets from the paper measurements. 87
5.7 The experimental setup for image-based BRDF measurement. 88
5.8 The camera path for an actual measurement. 88
5.9 The configuration of filters used in front of the primary camera. 89
5.10 The experimental setup for image-based BRDF measurement. 89
5.11 The BRDF samples generated by image-based measurement of the blue paint. 90
5.12 The sampling pattern from Figure 5.11 projected down to the disc. 91
5.13 Photographs of the actual test samples used. 91
5.14 Incidence plane measurements of the BRDF of paper. 92
5.15 Incidence plane measurements of the BRDF of paper. 93
5.16 Reciprocity comparison. 93
5.17 Gonioreflectometer comparison for office paper. 94
5.18 Incidence plane BRDF measurements for the gray primer. 95
5.19 Incidence plane BRDF measurements for the gray primer. 96
5.20 Reciprocity comparison for gray primer. 97
5.21 BRDF measurements of the gray primer at various incidence angles. 98
5.22 BRDF measurements of the blue paint through the red filter. 99
5.23 BRDF measurements of the blue paint through the green filter. 100
5.24 BRDF measurements of the blue paint through the blue filter. 101
5.25 BRDF measurements of the red paint through the red filter. 102
5.26 BRDF measurements of the red paint through the green filter. 103
5.27 BRDF measurements of the red paint through the blue filter. 104
5.28 The image-based measurements of the blue paint with the corresponding measurements from the gonioreflectometer. 105
5.29 The image-based measurements of the red paint with the corresponding measurements from the gonioreflectometer. 106
5.30 Cylindrical coordinates. 108
5.31 The cylinder corresponding to the mapping ψ_1. 108
5.32 The relationship between ψ_1 and ψ_2. 111
5.33 The sphere corresponding to the mapping ψ_2. 112
5.34 The cone corresponding to the mapping ψ_3. 113

5.35 Reconstructing a 1D function from irregular samples in four different ways. 115

A.1 The pinhole camera model. 122
A.2 Correcting for lens falloff in flat-field calibration. 125

B.1 Notation for bundle adjustment. 128

C.1 The meaning of the ID code printed around a target. 130
C.2 Sampling and decoding a noisy target image. 132

D.1 Spectral transmittance of filters used with PXL camera. 136
D.2 The spectral energy distribution of the SB-16 flash. 138

Chapter 1

Introduction

For much of its history, creating realistic images has been a major focus in the study of computer graphics. This effort was originally motivated by a desire for *photorealistic* images, which reproduce all aspects of an image necessary to make it indistinguishable from a photograph. More recently, many researchers have pursued the stronger goal of *predictive*, or *physically realistic*, images, which accurately describe the physical quantities that would be measured from a real scene. Significant progress has been made toward the goal of physical accuracy: today we have well-grounded mathematical descriptions of light reflection and light transport, and many algorithms are capable of rendering physically accurate images for a variety of scenes.

All realistic rendering is based, one way or another, on the *rendering equation:*

$$f = \mathbf{K}(h + \mathbf{G}f). \tag{1.1}$$

This equation, which follows Arvo's formulation [6], is derived in Chapter 2. It relates the rendered image to the characteristics of the scene being rendered as follows:

$$\underbrace{f}_{\substack{\text{Light}\\\text{reflected}\\\text{from}\\\text{surfaces}}} = \underbrace{\mathbf{K}}_{\substack{\text{How surfaces}\\\text{reflect light}}} \Big(\underbrace{h}_{\substack{\text{Direct}\\\text{illumination}\\\text{from light}\\\text{sources}}} + \underbrace{\underbrace{\mathbf{G}}_{\substack{\text{How light}\\\text{travels}\\\text{among}\\\text{surfaces}}} \underbrace{f}_{\substack{\text{Light reflected}\\\text{from surfaces}}}}_{\text{Indirect illumination}} \Big)$$

- The function f describes the light reflected from the surfaces in the scene, including light leaving every part of the surface in every direction.

- The operator $\mathbf{K}$ represents the reflectance of the surface, describing how it varies with surface position and with the directions of illumination and reflection.

- The operator $\mathbf{G}$ describes how light travels from one surface point to another, and depends solely on the geometry of the surfaces.
- The function h describes the incident light at each surface point due to direct illumination.

When a camera takes a photograph of a scene, the image measures light reflected from visible surfaces toward the camera—this is part of f. For this reason, a renderer must solve Equation 1.1 for f, given known values for the three other quantities. However, it is also possible, given knowledge of f, to pose *inverse rendering* problems in which one of the other quantities becomes the unknown. For input to these problems, we can use the information about f that we obtain from photographs taken of real objects. Because the mathematical models and algorithms we use for rendering are physically accurate, we can use the existing set of techniques that have been developed for rendering as we design algorithms to solve inverse rendering problems.

If h is unknown and $\mathbf{K}$, $\mathbf{G}$, and part of f are known, we have the problem of *inverse lighting*: given a photograph (part of f) and a complete model of the scene ($\mathbf{K}$ and $\mathbf{G}$), deduce what lighting (h) illuminated the scene. This problem is discussed in Chapter 3, and a solution technique is presented. We also demonstrate an application of inverse lighting to modifying lighting in photographs.

If $\mathbf{K}$ is unknown and $\mathbf{G}$, h, and part of f are known, we can solve for some information about $\mathbf{K}$. This problem, *image-based reflectometry*, is the topic of Chapters 4 and 5. Because $\mathbf{K}$ includes information about both spatial and directional variation in reflectance, it is potentially a very complicated function, and we will use a number of photographs, rather than just one, to record enough information. Even so, we must make simplifying assumptions in order to have tractable problems. In Chapter 4, we put constraints on the directional variation so that we can capture the spatial variation in detail—this is *photographic texture measurement*. In Chapter 5 we assume spatial uniformity so that we can capture the directional variation in detail—this is *image-based BRDF measurement*.

If $\mathbf{G}$ is unknown, we have the problem of *shape from shading*, which is a long-studied problem in the field of computer vision. We do not address shape from shading in this dissertation.

In addition to the three chapters of core material just mentioned, Chapter 2 covers some background ideas that are common to all the later chapters, and the Appendices describe some apparatus and methods that were important in the implementation of the techniques described in the chapters.

Chapter 2

Background

The techniques presented in the later chapters share a common theme of using computations from rendering. Radiometry provides the mathematical language of rendering; in particular, describing surface reflectance is an important topic with many subtleties. Using these ideas, we can formulate the rendering equation, which ultimately underlies all rendering and inverse rendering.

2.1 Mathematical Preliminaries

When we write integrals, we will use the notation common in measure theory [27], in which a function f is integrated over a domain A with respect to a *measure* μ. The resulting integral will be written

$$\int_A f \, d\mu \quad \text{or} \quad \int_A f(x) \, d\mu(x). \tag{2.1}$$

The latter form is used whenever it is important to name the integration variable. The measure serves to define the notion of size for subsets of the domain: for example, volume is a measure on 3-space, and surface area is a measure on the unit sphere.

The sphere, denoted S^2, will often represent the set of all directions in 3-space. When we are discussing the directions from a point on a surface, we will often separate the directions pointing inward from those pointing outward, splitting the sphere of directions into two hemispheres. The outward-facing hemisphere at a surface point $\mathbf{x}$ will be denoted $\Omega_{\mathbf{x}}$. Formally,

$$\Omega_{\mathbf{x}} = \{\omega \in S^2 \mid \langle \mathbf{n}(\mathbf{x}), \omega \rangle > 0\}, \tag{2.2}$$

where $\mathbf{n}(\mathbf{x})$ is the surface normal at $\mathbf{x}$. The inward-facing hemisphere is simply $-\Omega_{\mathbf{x}}$. When it is not necessary to make the dependence of $\Omega_{\mathbf{x}}$ on $\mathbf{x}$ explicit, the $\mathbf{x}$ will be omitted. We

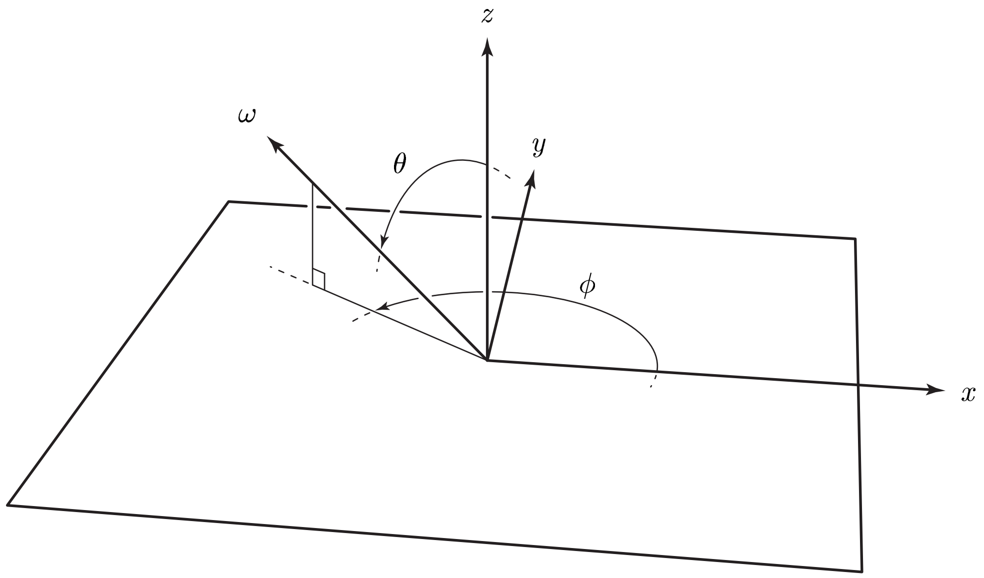


Figure 2.1: Coordinate system for directions at a surface.

will use unit vectors or spherical coordinates for elements of S^2 or Ω as needed, using the coordinate system shown in Figure 2.1.

Two measures on the hemisphere will be used: σ is the Lebesgue measure, which corresponds to the natural idea of spherical surface area, or solid angle; and μ is the projected solid angle measure. If A is a measurable subset of Ω, $\sigma(A)$ is the surface area of A (or equivalently the solid angle subtended by A) and $\mu(A)$ is the area of A projected onto the plane of the surface (Figure 2.2). The measures σ and μ are related by:

$$\int_A f\, d\mu = \int_A f(\omega)\, \langle \omega, \mathbf{n} \rangle\, d\sigma(\omega), \tag{2.3}$$

so μ is sometimes called the cosine-weighted solid angle measure.

A few notational conventions that may be unfamiliar can be defined by example:

- The symbol $\mathbf{x}$ normally represents a vector, and its components are $x_1, \ldots, x_n$.

- If $f(x,y) = xy$ is a function of two real variables, we can define f by writing:

$$\begin{aligned} f &: \mathbb{R}^2 \to \mathbb{R} \\ &: (x,y) \mapsto xy. \end{aligned}$$

- If $f : \mathbb{R}^2 \to \mathbb{R}$, then

$$\begin{aligned} f(x,\cdot) &: \mathbb{R} \to \mathbb{R} \\ &: y \mapsto f(x,y). \end{aligned}$$

- If $\sim$ is an equivalence relation on the set U, then $U\ /\sim$ is the *quotient* of U by $\sim$, which is the set of all equivalence classes of $\sim$. It can be thought of as a set obtained by identifying any pair of equivalent points to be the same point. For example, if we take U to be the unit square in $\mathbb{R}^2$ and define $\mathbf{x} \sim \mathbf{y} \iff x_1 = y_1$, then the equivalence classes of $\sim$ are all the vertical segments through the square, and $U\ /\sim$ amounts to a single horizontal segment. If U has a topological structure, then it induces a *quotient topology* on $U\ /\sim$ [43].

2.2 Radiometry

Rendering and inverse rendering are concerned with the propagation of light, which is described by the science of *radiometry.* We will formulate radiometry along much the same lines as Arvo [6].

We start with two assumptions: that light can be completely described by geometric optics and that all light is unpolarized. The former assumption leads to two properties that we will use extensively: light travels only along straight lines, and the effects of light sources are linear. In this case, linear means that if two sources of light A and B are used to illuminate some optical system, then the output from that system due to $A + B$ is the sum of the outputs due to A and B individually.

The most fundamental radiometric quantity is *radiance*, which completely describes the flow of light through space. The radiance function $L : \mathbb{R}^3 \times S^2 \to \mathbb{R}$ is defined for all directions at all points in space. $L(\mathbf{x}, \omega)$ measures the flow dP of light energy that would cross a small surface $d\mathbf{x}$ located at $\mathbf{x}$ facing the direction ω, counting only light traveling in a small range of directions $d\omega$ around ω (Figure 2.3), in proportion to the area of $d\mathbf{x}$ and the solid angle of $d\omega$:

$$dP = L(\mathbf{x}, \omega)\, d\mathbf{x}\, d\omega; \tag{2.4}$$

or,

$$L(\mathbf{x}, \omega) = \frac{\partial P}{\partial \mathbf{x}\, \partial \omega}. \tag{2.5}$$

The units of radiance are watts per square meter per steradian. It is important to note that radiance describes the flow through a surface when the surface is *perpendicular* to the direction of propagation. The flow per unit area and solid angle through a surface oriented with a surface normal $\mathbf{n}$ is

$$\frac{\partial P}{\partial \mathbf{x}\, \partial \omega} = L(\mathbf{x}, \omega) \left\langle \mathbf{n}, \omega \right\rangle . \tag{2.6}$$

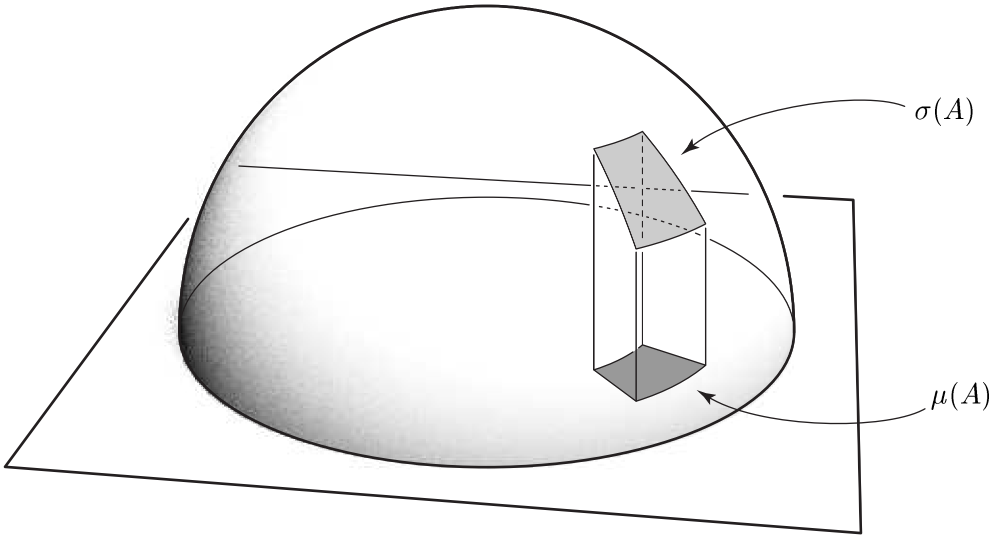


Figure 2.2: The solid angle measure σ and the projected solid angle measure μ.

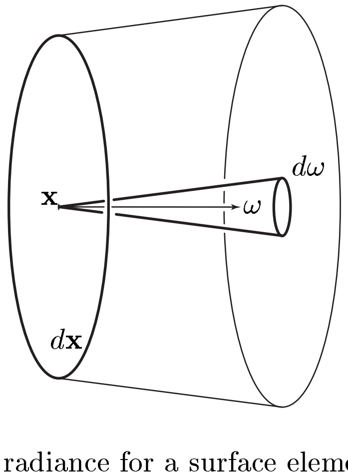


Figure 2.3: Measuring radiance for a surface element $d\mathbf{x}$ and a solid angle element $d\omega$. Only light that flows through $d\mathbf{x}$, the left face of the illustrated solid, traveling in directions within the cone $d\omega$, is included.

At a surface it is often convenient to separate the function $L(\mathbf{x},\cdot)$ into two functions $L_i(\mathbf{x},\cdot)$ and $L_e(\mathbf{x},\cdot) : \Omega \to \mathbb{R}$. The function $L_i(\mathbf{x},\cdot)$ represents only radiance traveling toward the surface, and $L_e(\mathbf{x},\cdot)$ measures only radiance traveling away from the surface:

$$L_i(\mathbf{x},\omega) = L(\mathbf{x},-\omega); \qquad L_e(\mathbf{x},\omega) = L(\mathbf{x},\omega). \tag{2.7}$$

Again, where the dependence on $\mathbf{x}$ need not be mentioned, the $\mathbf{x}$ will be omitted.

Several other radiometric quantities may be derived by integrating radiance. Irradiance is a measure of total power falling on a surface per unit area: $E(\mathbf{x}) = dP/d\mathbf{x}$. This total can be found by integrating Equation 2.6 with respect to solid angle:

$$E(\mathbf{x}) = \int_\Omega L_i(\mathbf{x},\omega)\,\langle \mathbf{n},\omega\rangle\; d\sigma(\omega). \tag{2.8}$$

Using Equation 2.3, we can simplify this to

$$E(\mathbf{x}) = \int_\Omega L_i(\mathbf{x},\cdot)\, d\mu. \tag{2.9}$$

In this way, we measure incident directions so as to account for the foreshortening of the surface area, avoiding the need for extra factors of $\langle \mathbf{n},\omega\rangle$ that would otherwise appear in every equation.

The analog of irradiance for exitant radiance is *radiant exitance*, a measure of the total power leaving a surface per unit area:

$$R(\mathbf{x}) = \int_\Omega L_e(\mathbf{x},\cdot)\, d\mu. \tag{2.10}$$

Both irradiance and radiant exitance have units of watts per square meter.

All these radiometric quantities depend on the wavelength of light. Radiance exists independently at every wavelength, and its complete description is a function of one more dimension:

$$L : \Lambda \times \mathbb{R}^3 \times S^2 \to \mathbb{R}; \tag{2.11}$$

$$L(\lambda,\mathbf{x},\omega) = \frac{\partial P}{\partial\lambda\,\partial\mathbf{x}\,\partial\omega}. \tag{2.12}$$

In this case L is called *spectral radiance* and has units of watts per square meter per steradian per nanometer. Wavelength dependence is important in computer graphics because variations in L with λ give rise to the sensation of color. The dependence of radiometric quantities on wavelength will be left implicit through most of this dissertation, since it rarely impacts the algorithms being developed. Where it is necessary, we will incorporate the wavelength dependence by measuring total radiance in each of several wavelength bands. Transforming such measurements into estimates of spectra or into colors that can be reproduced on a monitor or other output device is a challenging problem that we do not address.

2.3 The Bidirectional Reflectance Distribution Function

When the flow of light is interrupted by an opaque surface, the energy is partly absorbed by the material and partly scattered back toward the rest of the environment. The amount of scattered light and its directional distribution depend on the composition and structure of the surface.

2.3.1 Definition and basic properties

The easiest way to characterize how a surface scatters light is to describe how light arriving at a point from an infinitesimal solid angle is scattered into all directions. Once we have that information, we can integrate with respect to incoming solid angle to discover how light will scatter from any incident distribution.

The *bidirectional reflectance distribution function* [45], universally abbreviated BRDF, describes surface reflection according to this approach. For any incident direction ω_i and any exitant direction ω_e, the BRDF value $f_r(\omega_i, \omega_e)$ gives the ratio of radiance observed in the direction ω_e to irradiance from an infinitesimal solid angle about ω_i. In other words, it describes the result of the following experiment: expose the surface to a uniform radiance of L_i coming from a small solid angle Ω_i containing ω_i. Measure the radiance L_e reflected from the surface in the direction ω_e for various sizes of Ω_i. The ratio L_e/L_i will be directly proportional to $\mu(\Omega_i)$,[1] and the constant of proportionality is the BRDF, $f_r(\omega_i, \omega_e)$, which therefore has units of inverse solid angle. Knowing the exitant radiance per unit incident radiance per unit solid angle for any incident direction, we can integrate with respect to solid angle to find the exitant radiance due to an entire incident radiance distribution:

$$L_e(\omega_e) = \int_\Omega f_r(\omega_i, \omega_e) L_i(\omega_i)\, d\mu(\omega_i). \tag{2.13}$$

A BRDF is then a function $f_r : \Omega \times \Omega \to \mathbb{R}$; it is a real-valued function on a four-dimensional domain. Another way to describe a surface's directional reflectance behavior is with a linear operator $\mathbf{F}_r$ that maps a real-valued function on Ω (the incident radiance distribution) to another real-valued function on Ω (the reflected radiance distribution). The action of $\mathbf{F}_r$ is defined by Equation 2.13, so that $L_e = \mathbf{F}_r L_i$.

All BRDFs share two properties that are required by physical law. First, the BRDF of any surface is always symmetric with respect to exchanging its arguments:

$$f_r(\omega, \xi) = f_r(\xi, \omega). \tag{2.14}$$

[1] This is exactly true only in the limit as $\mu(\Omega_i) \to 0$.

This property is called *reciprocity*, and it has the effect of reducing the domain of f_r by a factor of two. If we define the equivalence relation $\sim_r$ such that $(\omega, \xi) \sim_r (\xi, \omega)$ for all directions ω and ξ, then reciprocity means that two configurations that are equivalent under $\sim_r$ map to the same value under f_r. By requiring f_r to agree on equivalent configurations, we have in effect defined BRDFs to be functions on $(\Omega \times \Omega) \,/\, \sim_r$. Although this domain is in some sense smaller than $\Omega \times \Omega$, it continues to be a four-dimensional set.

For the linear operator formulation of the BRDF, reciprocity simply requires $\mathbf{F}_r$ to be self-adjoint.

All BRDFs also have the property that the reflected radiant exitance is never greater than the incident irradiance:

$$\int_\Omega \int_\Omega f_r(\omega_i, \omega_e) L_i(\omega_i)\, d\mu(\omega_i) d\mu(\omega_e) < \int_\Omega L_i(\omega)\, d\mu(\omega) \quad \text{for all } L_i. \tag{2.15}$$

Since this bound must hold for any incident distribution L_i, it must hold for a distribution with all its energy in an infinitesimal solid angle around any particular direction ω_i, so that

$$\int_\Omega f_r(\omega_i, \omega_e)\, d\mu(\omega_e) < 1 \quad \text{for all } \omega_i. \tag{2.16}$$

Conversely, if we assume Equation 2.16, then integrating both sides against L_i over Ω leads to Equation 2.15, so the two equations are equivalent. We will therefore use the simpler Equation 2.16 as the definition of the property of energy conservation.

For the linear operator $\mathbf{F}_r$, energy conservation means that the 1-norm of $\mathbf{F}_r$ is less than 1.

Another BRDF property, one shared by many materials, is *isotropy*. An isotropic surface is one that has no "grain" or distinguished direction to its material. Such a surface has a BRDF that is invariant under rotating the surface in its plane. That is,

$$f_r(\omega_i, \omega_e) = f_r(R\omega_i, R\omega_e), \tag{2.17}$$

where R is any rotation about the surface normal (Figure 2.4). This defines an equivalence relation on $\Omega \times \Omega$ under which $(\omega, \xi) \sim_i (R\omega, R\xi)$ for all directions ω and ξ and all rotations R about the surface normal. The equivalence classes of this relation are one-dimensional subsets of $\Omega \times \Omega$: each consists of all the configurations that can be obtained by rotating a single configuration around the surface normal. The effective domain of an isotropic BRDF is then $(\Omega \times \Omega) \,/\, \sim_i$, which is a three-dimensional set. Combining this domain reduction with the reduction due to reciprocity leads to a further reduced but still three-dimensional domain, $(\Omega \times \Omega) \,/ (\sim_r \cup \sim_i)$.[2]

[2]Two configurations are equivalent under $\sim_r \cup \sim_i$ if they are equivalent under either $\sim_r$ or $\sim_i$.

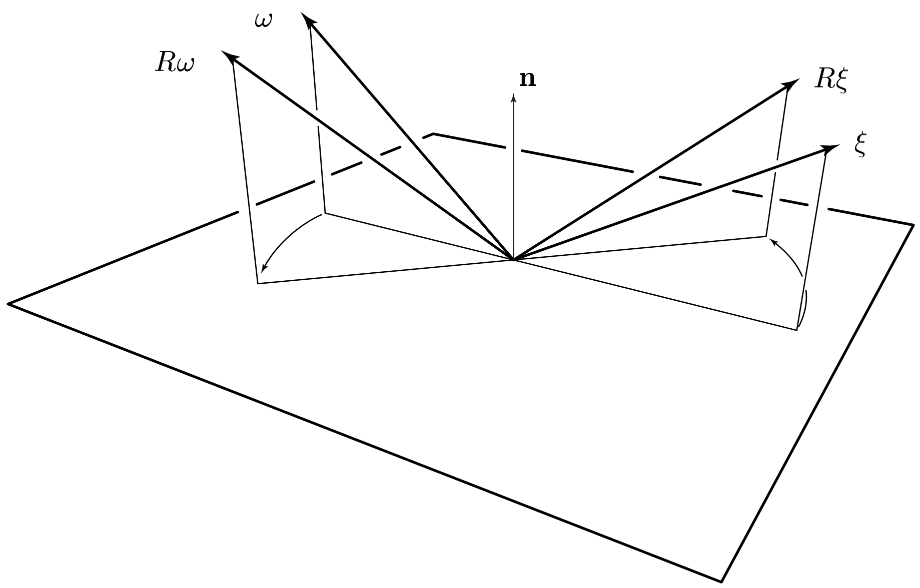


Figure 2.4: Rotating a configuration about the surface normal. For an isotropic BRDF, $f_r(\omega, \xi) = f_r(R\omega, R\xi)$.

A reasonable extension to the concept of isotropy is *bilateral symmetry*: the exitant distribution for any particular incident direction should be symmetric across the plane of incidence. This symmetry does not follow from reciprocity and isotropy, but it is intuitively reasonable that a surface without any directionally aligned behavior will also fail to distinguish between scattering to the left and to the right. If this symmetry is assumed, it leads to another halving of the domain of the BRDF.

2.3.2 Common reflectance phenomena

The previous section outlined the fundamental properties of the BRDF, but the BRDFs of most materials have many other characteristics in common. There are a few modes of reflection that account for the most commonly observed features of measured BRDFs. These behaviors are related to the fine geometric structure of the surface.

Specular reflection occurs only for smooth surfaces. In this mode of reflection, light coming from one direction is reflected into a single outgoing direction. The incident distribution of radiance is exactly duplicated, though attenuated, in the exitant distribution, but reflected through the normal. Information about the incident light distribution is not lost, and an optical image of the source is formed. Examples of surfaces that exhibit exclusively specular reflection are a front-silvered mirror, which reflects nearly all the incident light, and a smooth piece of glass, which reflects a small portion of the incident light, transmitting the rest.

In *diffuse reflection*, incoming light from a single direction is scattered into a range of directions. The distribution of reflected light from a narrow incident beam differs in shape depending on the characteristics of the surface. In the purest form of diffuse reflection, the radiance distribution approaches uniform, in the limit becoming an *ideal diffuse*, or *Lambertian*, reflector; such a surface has a constant function for its BRDF.

Diffuse reflection that is not Lambertian is called *directional diffuse* [30]. Directional diffuse reflectance distributions for a particular incident direction typically have a peak near the specular direction, with radiance falling off away from that peak. As the angle between the incident direction and the surface normal increases, the reflectance peak exhibits two changes: it increases in magnitude, and the direction at which maximum reflectance is achieved moves away from the specular direction—the peak becomes increasingly *off-specular*. These two phenomena are often considered to be separate, but they are in fact two aspects of the same phenomenon. This can be seen by examining a graph of the incidence-plane values of an isotropic directional diffuse BRDF as a function of the incidence and exitance angles.

Figure 2.5 (a) shows such a plot for a simple isotropic BRDF model. The configurations where specular reflection would occur are along the diagonal of the graph, where $\theta_i = \theta_e$, and the graph is symmetric across that diagonal (because of reciprocity and isotropy)[3]. The directional peak of the BRDF is a ridge along this line, and because the reflectance increases with increasing incidence angle, the ridge gets higher as $|\theta_i|$ and $|\theta_e|$ increase (towards the back in the figure). The line on the surface where $\theta_i = \theta_e$ is drawn with a thin black line.

When we draw the exitant distribution for a particular angle of incidence, we are taking a slice of the surface in Figure 2.5 (a). One such slice, for an incident angle around 65°, is drawn with a bold black line, and it is repeated in part (b) as a standard plot of f_r versus θ_e. At the specular angle (the intersection of the black curves), the fixed-incidence curve cuts obliquely across the upward-sloping ridge, and is therefore sloping upwards.[4]

Therefore, we can conclude that the peak of a fixed-incidence reflectance distribution *cannot* be in the specular direction if the magnitude of the peak increases with angle. Off-specular reflection is a direct consequence of increasing reflectance toward grazing angles, and it is really an artifact of how we slice the BRDF when we graph fixed-incidence distributions. If the peak in the directional diffuse reflection is considered as a characteristic of the whole BRDF, rather than as a characteristic of the slices of the BRDF, then it is centered on the specular diagonal, as can be seen from Figure 2.5, and is not off-specular

[3]To be precise, $\theta_i = -\theta_e$, but for simplicity in this section both angles are positive.

[4]A brief proof: the directional derivative across the ridge, perpendicular to the line of symmetry, is zero, and if the directional derivative along the fixed-incidence slice was also zero (as it would be if the peak of the fixed-incidence distribution was at the specular angle) the entire derivative would be zero, contradicting the fact that the ridge is sloping upwards.

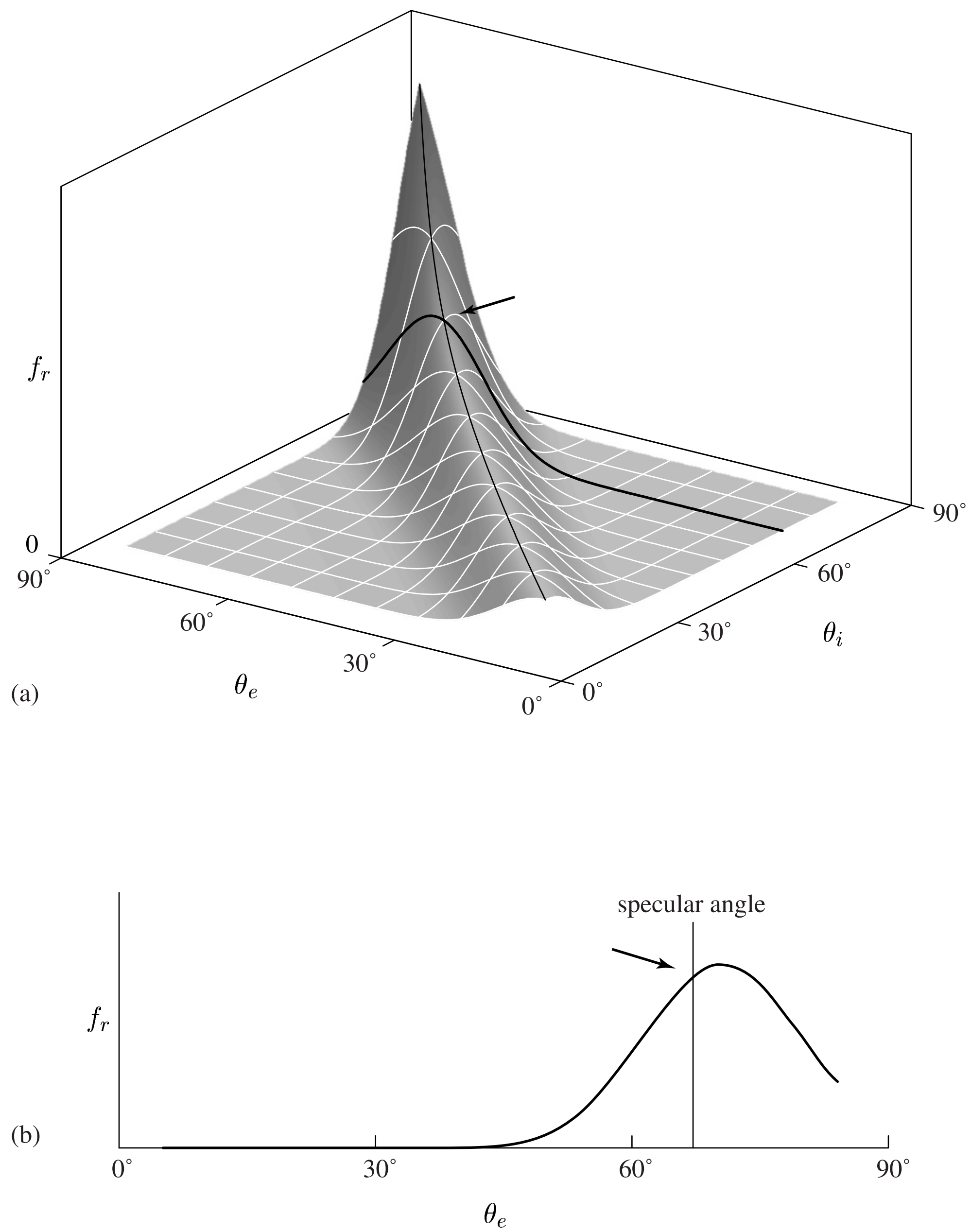


Figure 2.5: The relationship between off-specular reflection and increasing specularity. (a) The BRDF in the incidence plane plotted as a function of incidence and exitance angle. The two curves drawn in black are the ridge line (light), where $\theta_i = \theta_e$, and a slice for fixed incidence angle (bold). (b) The same slice plotted as a function of incidence angle alone. Note that in both graphs the bold curve is still increasing where the two curves cross (arrow).

at all.

An additional reflectance phenomenon that often occurs in conjunction with diffuse reflection is known as *retroreflection* (sometimes called the *hot spot* effect [3]). In this mode of reflection, which is common among rough and porous surfaces like cloth, stone, and flat paints, the scattered light distribution shows a peak pointing back in the direction of the source. The most dramatic examples of retroreflection come from materials designed for use in traffic signs and other objects that must be visible to drivers of automobiles at night. When they are illuminated by a light source (the car's headlights) positioned near the viewer, they return a large fraction of the incident light back toward the viewer, making them appear many times brighter than other surfaces. Most examples of such extreme retroreflection are man-made surfaces, but the same phenomenon occurs to a lesser extent in a great variety of materials.

All measurements of BRDFs must be made over some surface area at least large enough to enable the assumptions of geometric optics. How large an area is used affects which of these types of reflection will be observed, because the scale of measurement determines which surface features are treated as microstructure to be rolled into the BRDF and which are treated as part of the geometry of the surface on which the BRDF is being measured. For example, suppose we are measuring an irregular stone surface that has been painted with a glossy paint. If we measure one square millimeter of the surface, we will conclude that we have a specular surface, since the paint will have filled in all the surface imperfections to produce a locally smooth surface. However, if we measure a square meter, we will find that light is scattered in many directions from the differently oriented parts of the surface, leading to a directional diffuse reflectance distribution. When measuring real surfaces, particularly those (such as the objects measured in Chapter 4) that have complex shapes, it is always important to make sure that surface detail that is not represented by the geometric model is properly accounted for in the BRDF but that geometry that is modeled is not included in the BRDF.

2.4 Rendering

One reason for studying radiometry and BRDFs is to simulate the reflection of light in order to produce realistic synthetic images of modeled scenes. This problem has been studied extensively, and many techniques for solving it have been described. Many of the computations involved in those algorithms are the same ones that will be required to solve the inverse problems discussed in this dissertation.

The BRDF describes how light reflects at a single surface point in isolation. Describing how light moves around an entire environment requires connecting the surface BRDFs with a description of how light gets from surface to surface. The equation that does this is called the *rendering equation*, and we develop it here using a formulation similar to Arvo's [6].

We will consider the environment to consist of various light-reflecting surfaces illuminated by light that originates outside the convex hull of all surfaces; we refer to illumination as coming from the *background*. This background, along with the assumption that surfaces do not emit light, is a departure from the common practice in the rendering literature, but in the subsequent chapters it will prove more convenient than the usual formulation.

Let $\mathcal{M}$ be a piecewise-smooth 2-manifold embedded in $\mathbb{R}^3$, which describes the shape of all the surfaces in the environment. We extend the function f_r with an additional parameter that makes the dependence on surface position explicit:

$$f_r : \mathcal{M} \times \Omega \times \Omega \to \mathbb{R}. \tag{2.18}$$

Lastly, we define a function $L_i^0 : \mathcal{M} \times \Omega \to \mathbb{R}$ that gives the incident radiance due directly to illumination from the background in every direction at every surface point.

We define two linear operators on the space of real-valued functions on $\mathcal{M} \times \Omega$ that encapsulate how light behaves. First, we encapsulate f_r in the *reflection operator* $\mathbf{K}$:

$$(\mathbf{K}L_i)(\mathbf{x}, \omega_e) = \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_e) L_i(\mathbf{x}, \omega_i)\, d\mu(\omega_i). \tag{2.19}$$

$\mathbf{K}$ transforms an incident light distribution into the exitant light distribution that results by reflection. Second, we define the transport operator $\mathbf{G}$:

$$(\mathbf{G}L_e)(\mathbf{x}, \omega_i) = \begin{cases} L_e(\mathbf{y}, -\omega_i) & \text{if } \mathbf{y} \text{ is visible from } \mathbf{x} \text{ in the direction } \omega, \\ 0 & \text{if no surface is visible.} \end{cases} \tag{2.20}$$

$\mathbf{G}$ transforms an exitant light distribution into the incident light distribution that results from surfaces illuminating one another.

These two operators lead to a compact form of the rendering equation. The inputs to the rendering process are the geometry, which makes $\mathbf{G}$ known; the BRDFs, which make $\mathbf{K}$ known; and the lighting, which makes L_i^0 known. $\mathbf{K}$ lets us write L_e, for which we must solve, in terms of L_i:

$$L_e = \mathbf{K}L_i. \tag{2.21}$$

In turn, L_i is the sum of direct illumination, L_i^0, and illumination by light exiting other surface points, $\mathbf{G}L_e$. Thus,

$$L_e = \mathbf{K}(L_i^0 + \mathbf{G}L_e). \tag{2.22}$$

This equation has a single unknown, L_e, and it is an operator equation of the second kind [6]. Methods for solving this equation, the rendering equation, have been studied extensively, and many effective algorithms have been found. The method we used to render images in this dissertation, both inside algorithms and for presentation, is Monte Carlo path tracing [35, 5, 56, 24].

The only difference between Equation 2.22 and the rendering equation presented by Arvo is how light gets into the system. He uses a surface emission function L_e^0 and writes $L_e = \mathbf{KG} L_e + L_e^0$. In our introduction, we wrote f for L_e and h for L_i^0 in order to be more consistent with his notation.

Chapter 3

Inverse Lighting

For as long as photographs have existed, photographers have sought ways to improve their images after they have been taken. Professional photographers have perfected the analog techniques for developing and printing film with a variety of effects, such as enhancing contrast or raising or lowering the luminance of the entire scene. They commonly manipulate individual parts of a photograph, removing shadows, highlighting features, softening backgrounds, and sharpening detail by optically focusing portions of the image in different ways or exposing regions of the picture in different amounts. These time-consuming manual operations, which yield impressive photographic results, require great artistic and technical skill.

Many of these same operations are now standardly performed on digital images using digital filters that are convolved with the array of intensity values. Filters have been designed for blurring, sharpening, edge detection, contrast enhancement, and many other operations. Formerly used only in the scientific image processing domain, these operations are now widely available through commercial software programs, such as Adobe Photoshop [1]. These easy-to-use programs all assume a two-dimensional array of input values and provide two-dimensional operations that result in two-dimensional output arrays. Although results can be impressive, some tasks, such as removing shadows, changing lighting conditions, or modifying the shading on continuous surfaces, are difficult to achieve.

With the advent of digital photography, it has become possible for cameras to record with each photograph information external to the actual image. For instance, many digital cameras record the lens settings, exposure parameters, distance to subject, and other data available from the computer that controls the camera's operation. In the future, more sophisticated cameras will gather more complex information about the scene. What if this included geometry? What additional operations would be possible that are impossible without such information?

This question leads us to the first inverse rendering problem we will consider. As we pointed out in the introduction, having a photograph and a description of the subject's geometry and reflectance allows us to pose the *inverse lighting* problem, which asks for a description of the sources of light that illuminated the object when the photograph was taken. In this chapter, we will pose this problem more precisely, then proceed to solve it using methods for linear inverse systems. We will also demonstrate an application of the resulting data, which will allow us to manipulate the lighting in the photograph.

3.1 Problem Statement

Inverse lighting is an inverse rendering problem in the most literal sense. We are given a description of a scene and of a camera viewing that scene—that is, all the usual inputs to a renderer other than the lighting. We are also given a photograph of the described scene taken by the described camera—that is, the usual output of a renderer. Our task is to determine the lighting that was used to take the photograph, or, more literally, what lighting could be provided as input to a renderer to cause an output similar to the photograph.

For this problem statement to be precise, we must define exactly what we mean by asking what the "lighting" is in the photograph. In general, light could originate anywhere, even directly from the surfaces of the objects pictured in the photograph. With this possibility, it is impossible to distinguish an object that glows from one that is reflecting light, but excluding self-emissive objects will not significantly harm the practical applicability of our algorithm. However, even if light is assumed to come from out of view, a four-parameter function is still required to completely describe the incoming radiance field, since a different radiance could conceivably arrive from every line entering the scene [39]. We cannot recover that function from a single two-parameter image.

We chose to reduce the dimension of the lighting solution by assuming that all light comes from sources that are far away compared to the size of the objects we are looking at. This means that the incident radiance distribution is the same[1] at every point on the object's surface. We can think of this radiance as coming from the inside of a very large sphere surrounding the scene. Thus a light distribution is a two-parameter function $L_b : S^2 \to \mathbb{R}$. In essence, we are solving for the function L_i^0 of Section 2.4 under the assumption that it does not vary with position on the surface.

3.2 Prior Work

Solving for lighting in an inverse system is not new to computer graphics, although it may be new to photography. Schoeneman et al. introduced "painting with light," [55] in which a

[1] In a fixed frame of reference, rather than with respect to the local surface normal.

designer could specify the desired illumination at particular locations in a scene. Intensities of a fixed set of lights were adjusted by a linear least-squares procedure to approximate the desired result. Kawai et al. described the process of "radioptimization," [37] which uses a nonlinear minimization procedure with a more complex objective function to meet various design goals by adjusting light source directions and intensities.

In this chapter, we apply the idea of inverse lighting using a measured photograph, rather than a user-specified goal, and using a generic set of basis lights. This leads to a system that infers the lighting that actually existed at the time the photograph was taken, rather than solving for a lighting setup that achieves a desired effect. The system that results from using a generic set of lights to illuminate an object is more ill-conditioned than the system that results from lighting an environment with a set of fairly focused lights, requiring the use of a regularized least-squares procedure.

Others have taken advantage of the linearity of rendering in other ways. Nimeroff et al. rendered basis images illuminated by a set of steerable basis functions and used linear combinations to approximate the effects of skylight at different times of day [46]. Steerable basis functions were also used to allow efficient re-rendering under altered lighting in Teo et al.'s work on interactive lighting design [57]. Further examples include Dorsey et al.'s work on simulation for lighting design in opera [17] and Airey et al.'s work on real time building walkthroughs [4].

3.3 Basic Least-squares Solution

As explained in Section 2.4, the rendering problem for a particular scene defines a linear function from the background radiance distribution L_b, which we call lighting here, to the radiance reflected from the surface, L_e. This property corresponds to the familiar notion of superposition of light: if you light an object with two sources together, the resulting image will be the sum of the images that would result from each source separately. The linearity of rendering with respect to lighting has great significance, because it means that the large body of robust, efficient computational tools that exists for linear problems may be brought to bear on inverse lighting.

We can encapsulate the action of the rendering process on a lighting configuration in a single linear operator $\mathbf{R}$, which maps lighting configurations to reflected radiance distributions:

$$\mathbf{R} : (S^2 \to \mathbb{R}) \to (\mathcal{M} \times \Omega \to \mathbb{R}). \tag{3.1}$$

In this formulation, the information about the scene, both geometry and reflectance, is folded into $\mathbf{R}$.

In order to work with $\mathbf{R}$ in computation, we must represent its input, L_b, and its output, L_e, numerically. The information that we have about L_e is already in usable form: each pixel of the image is a linear measurement of reflected radiance over a small area of surface and a small cone of directions. Let $K_j : (\mathcal{M} \times \Omega \to \mathrm{I\!R}) \to \mathrm{I\!R}$ be the linear functional that gives the value of pixel j in terms of L_e. To represent a lighting configuration numerically, we approximate it by a linear combination of a set of basis functions $L_b^1, \ldots, L_b^n$. A lighting configuration is then specified by a set of coefficients $x_1, \ldots, x_n$, and the corresponding light distribution is $L_b = \sum_{i=1}^n x_i L_b^i$.

Let $b_1, \ldots, b_m$ be the values of the pixels in the photograph. Using the notation we just developed, we can write down the relationship between b_j and L_b:

$$b_j = K_j \mathbf{R} L_b. \tag{3.2}$$

Substituting the representation of L_b in our light basis, we have

$$b_j = K_j \mathbf{R} \left(\sum_{i=1}^n x_i L_b^i \right) = \sum_{i=1}^n x_i K_j \mathbf{R} L_b^i. \tag{3.3}$$

Since $K_j \mathbf{R} L_b^i$ is a scalar, this equation is a matrix system:

$$\begin{bmatrix} \vdots \\ b_j \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & K_j \mathbf{R} L_b^i & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ x_i \\ \vdots \end{bmatrix}. \tag{3.4}$$

If we let A represent the matrix, we have a compact description for our discrete approximation of the rendering process:

$$b = Ax. \tag{3.5}$$

The matrix A is m by n, where m is the number of pixels and n is the number of light basis functions. Note that column i of A simply contains the pixel values of an image of the object lit by the i^{th} basis function; we call this image the i^{th} *basis image*. This restating of the rendering process in terms of a light basis succinctly describes the image that will result from any light distribution described in terms of the basis: it is a weighted sum of the basis images, with the weight of each basis function becoming the weight of its corresponding basis image. It also provides a statement of the inverse lighting problem, if we make b the known value and x the unknown.

Since there are always more pixels in the image than there are light basis functions, $m > n$ and this system is overdetermined, and we may be unable to achieve equality, no matter what value we assign to x. We must relax our goal to finding a value for x that brings Ax close to b. Thus, solving inverse lighting amounts to finding a linear combination of the n basis images that closely resembles the photograph; the coefficients of this linear combination are then the coefficients of the light basis functions in the solution. Finding the *least-squares solution*, the value for x that brings Ax closest to b in the 2-norm, is a standard problem of numerical linear algebra that can be solved in time $O(n^2m)$ via any one of a number of methods [38, 49, 25].

The process of inverse lighting is summarized in Figure 3.1. First, the renderer is given the camera, the 3D model, and each of the basis lights, and it produces a set of basis images. The least-squares solution method then determines a linear combination of these basis images that matches the photograph. The coefficients in this linear combination are the lighting solution.

In order to use this technique, we must define the light basis functions $L_b^1, \ldots, L_b^n$. We chose to use a piecewise constant basis. We divided the sphere into a number of triangular regions and defined one basis function for each region. Each basis function is equal to one inside its corresponding region, called its *region of support*, and zero outside. The regions were defined as follows: start with the sphere divided along the coordinate planes into eight spherical triangles, then recursively subdivide each triangle into four smaller triangles to obtain bases of 32, 128, or 512 elements. Figure 3.2 illustrates the 32-element basis obtained by this process.

3.4 Regularized Solution

Particularly when the surface's BRDF is very smooth, for instance when a Lambertian BRDF is used, the matrix A is ill-conditioned. This means that A is nearly singular, so that some significant changes to x have little effect on Ax. In this situation, many values of x that are far from the correct x will bring Ax nearly as close to b as the correct answer does. With the introduction of noise and modeling errors, the least-squares value of x may no longer be anywhere near the answer we seek. The least-squares solution, while it does illuminate the object so as to create an image that closely approximates the photograph, is not reasonable as an estimate of the lighting in the actual scene. Even if the model and renderer were a perfect simulation of reality, so that the correct solution would exactly match the photograph, the ill-conditioning introduces enough sensitivity to noise that the noise inherent in capturing the photograph will lead the lighting solution to become wildly varying and implausible. In practice, the photograph will be somewhat different even from an image rendered using exactly the right lighting; this further compounds the problem.

In order to obtain solutions that are more plausible as real lighting configurations, we adopt a widely used technique known as *regularization*. We introduce a term $\mathcal{B}(x)$ into the quantity to be minimized that measures departure from "plausibility." Rather rather than

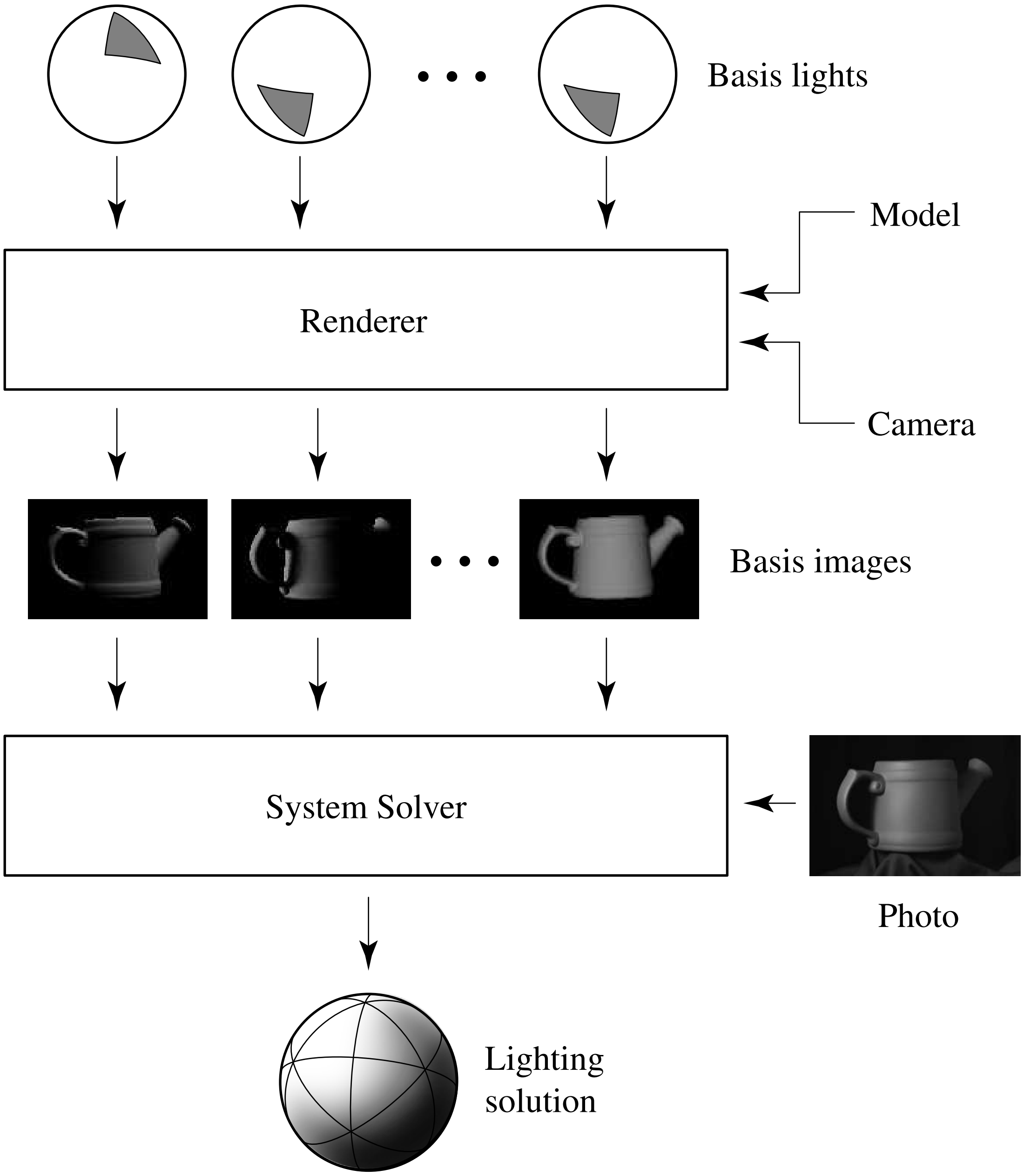


Figure 3.1: The data flow in the inverse lighting algorithm.

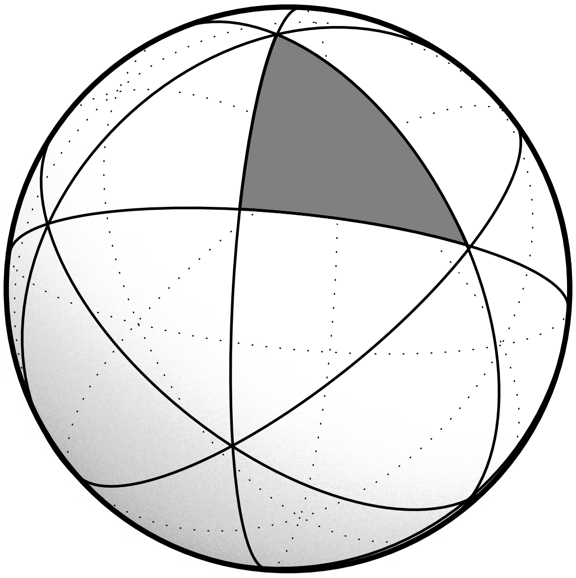


Figure 3.2: A 32-light basis set, drawn on the directional sphere.

minimizing $\|Ax - b\|^2$, we minimize $\|Ax - b\|^2 + \lambda^2 \mathcal{B}(x)$. If $\mathcal{B}$ is expressible as the squared norm of a linear function B of x, it can be combined with A to get a new linear system:

$$\begin{bmatrix} A \\ \lambda B \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{3.6}$$

This is called first-order linear regularization. The parameter λ mediates the tradeoff between how well Ax approximates b and how reasonable x is. In the absence of precise information about the errors in A and b, λ must be chosen by hand: this is the approach we take.

The choice of B of course affects the regularized solution. Since we are trying to eliminate unwarranted local variations in L_b, we chose to make $\mathcal{B}$ an approximation to the norm of the first derivative of L_b. This means that our task is to find a slowly varying radiance distribution that produces an image similar to the photograph. In our piecewise constant basis, this translates to penalizing differences in the coefficients of basis functions with adjacent regions of support. Thus, for every pair (L_b^i, L_b^j) of basis functions that are adjacent on the sphere, we add the square of the difference in their coefficients to $\mathcal{B}$:

$$\mathcal{B}(x) = \sum_{L_b^i, L_b^j \text{ adjacent}} (x_i - x_j)^2. \tag{3.7}$$

This definition leads to a matrix B that has a row for every pair of adjacent basis functions. Each row consists of all zeroes except for a one in the column of the first basis function in the pair and a negative one in the column of the second. Then $\|Bx\|^2 = \mathcal{B}(x)$.

3.4.1 Accelerating using the GSVD

We can easily compute x by forming the matrix in Equation 3.6 and using the same least-squares algorithm as in Section 3.3. However, this requires repeating the entire calculation for each new λ, which precludes interactive adjustment of λ. Instead, we use a method involving the generalized SVD (GSVD) that allows λ to be adjusted with just the cost of an n by n matrix-vector multiplication for each change in λ.

The GSVD [25] takes two matrices $A \in \mathbb{R}^{m_1 \times n}$ and $B \in \mathbb{R}^{m_2 \times n}$ and produces orthogonal matrices $U_1 \in \mathbb{R}^{m_1 \times m_1}$ and $U_2 \in \mathbb{R}^{m_2 \times m_2}$, an invertible matrix $X \in \mathbb{R}^{n \times n}$, and two diagonal matrices $C \in \mathbb{R}^{m_1 \times n}$ and $S \in \mathbb{R}^{m_2 \times n}$ such that

$$\begin{aligned} U_1^T A X &= C \\ U_2^T B X &= S. \end{aligned} \tag{3.8}$$

The advantage of factoring A and B with orthogonal factors on the left and a common factor on the right becomes apparent when we look at the normal equations for (3.6):

$$(A^T A + \lambda^2 B^T B)x = A^T b. \tag{3.9}$$

When we substitute the factored A and B, we obtain

$$(X^{-T} C^T U_1^T U_1 C X^{-1} + \lambda^2 X^{-T} S^T U_2^T U_2 S X^{-1})x = X^{-T} C^T U_1^T b. \tag{3.10}$$

Since U_1 and U_2 are orthogonal, $U_1^T U_1 = I$ and $U_2^T U_2 = I$. Canceling those and noting that $C^T = C$ and $S^T = S$ we have:

$$\begin{aligned} (X^{-T} C^2 X^{-1} + \lambda^2 X^{-T} S^2 X^{-1})x &= X^{-T} C U_1^T b \\ X^{-T}(C^2 + \lambda^2 S^2) X^{-1} x &= X^{-T} C U_1^T b \\ (C^2 + \lambda^2 S^2) X^{-1} x &= C U_1^T b \\ x &= X(C^2 + \lambda^2 S^2)^{-1} C U_1^T b. \end{aligned}$$

We have now factored the computation of x into two parts, an expensive part $CU_1^T b$ that does not depend on λ and an inexpensive part $X(C^2+\lambda^2 S^2)^{-1}$ that does. If we pre-compute the GSVD and the vector $y = CU_1^T b$, we can get a new x from a new λ by computing $(C^2 + \lambda^2 S^2)^{-1} y$ (an $O(n)$ operation, since C and S are diagonal) and multiplying by X (an $O(n^2)$ operation). The computation of the GSVD, if we compute only the first n columns of U_1 and U_2, is $O(n^2 m)$, the same order as computing x directly for a single value of λ. In practice, using this acceleration technique increases the time required to set up the system somewhat over direct computation, but computing each new value of x becomes essentially instantaneous.

3.5 Re-lighting

Once we have computed the existing lighting configuration, we can use that information to modify the lighting in the photograph. Again, we assume that we have a complete model of the object, including its reflectance, but we allow that the photograph contains fine detail that is missing from the model. That is, there are small-scale variations in the color of the surface that are visible in the photograph but not present in the model—these details are a large part of what makes a high-resolution photograph more realistic and appealing than a rendering of our model. We want to preserve these details while changing the lighting on the object.

The renderer, given a description of the existing and desired lighting, can predict the change in the image by producing two rendered images: one under the existing lighting (the "old rendering") and one under the desired lighting (the "new rendering"). The old rendering should look like the photograph but lack detail, and the new rendering shows generally how the modified photograph should look. We must then adjust the photograph to look like the new rendering, but do so without damaging the detail that distinguishes the photograph from the old rendering. In other words, we want to use the two renderings and the photograph to predict the desired photograph—the photograph that would have been taken under the desired lighting.

In the case of a diffuse surface, the radiance reflected at each point is directly proportional to the irradiance. This means that the ratio of the new to old radiance value at a pixel depends only on the ratio of new to old irradiance at the surface seen by that pixel, and not on its reflectance. Thus the ratio of the new to old rendered image, even if there is missing reflectance detail, correctly predicts the ratio of the desired photograph to the existing photograph.[2]

This observation that the renderer can correctly predict the ratio of new and old pixel values even under some deviations from the modeled surface reflectance leads to the idea of using such ratios in re-lighting. In particular, we set the ratio of the modified to existing photograph equal to the ratio of the new to old rendering. That is, we multiply the photograph, pixel by pixel, by the ratio of the new to old renderings, resulting in the modified photograph.

When computing the ratio of the two renderings, the values of individual pixels near the silhouette of the object can be very sensitive to noise in the computation. Noise in the rendered images resulted in isolated extreme pixels, and we found it helpful to run a 3-by-3 median filter over the ratio image to suppress these isolated outliers and reduce the noise.

To summarize, the process of re-lighting a photograph by using inverse lighting proceeds

[2]The surface need not be diffuse; this argument works as long as the real BRDF differs from the modeled BRDF only by a scale factor.

as follows (Figures 3.3 and 3.4). First, the previously described inverse lighting procedure is used to compute the existing lighting from the photograph, given the 3D model and the camera position and focal length. The user modifies this lighting solution to form a new, desired lighting configuration. The rendering system is then used in its normal forward mode to create the new and old renderings from the desired and existing lighting, again given the model and camera. Then the photograph is multiplied by the ratio of these renderings to produce the result, a modified photograph.

3.6 A Test with a Synthetic Photograph

We first demonstrate the system on a synthetic "photograph" to show how it works in the absence of errors in the model and the camera position. The model is a scan of a rabbit figurine,[3] and it consists of 6.9×10^4 triangles plus a rectangular floor that it sits on. The lighting configuration used to render the "original photograph" has two spherical light sources and a background that provides light from all directions. We modified the lighting configuration to be much more strongly directional, with light coming predominantly from above and to the viewer's left.

Figure 3.5 shows the results of this test. Part (a) shows the synthetic "photograph," and in (b) is the model rendered under the lighting solution found by inverse lighting. There are some subtle differences between the two—most notably, the shadow on the floor is less distinct and the left side of the rabbit's face is lit more strongly. The former effect is because the true light source can only be approximated by summing up the basis lights; the latter may be caused by the bright background contaminating some pixels on the model's silhouette. The same model rendered under the new lighting configuration is shown in part (c), and the ratio image, which is the ratio of (c) to (b), is shown in (d). The modified "photograph" is shown in part (e). Because this is a synthetic example in which the 3D model does exactly represent the scene, the intermediate renderings closely resemble the "photographs." A rendering of the model done directly under the new lighting configuration is shown in Figure 3.6 for comparison with the re-lit image. This rendering represents the correct answer that would be achieved by perfect re-lighting. Note that the intermediate images are computed at a lower resolution, and the ratio image was filtered using the technique specified in the previous section. This, along with the differences between the original "photograph" and the old rendered image, prevents the re-lit "photograph" from matching the comparison rendering exactly.

[3] Model courtesy of the Stanford 3D Scanning Repository.

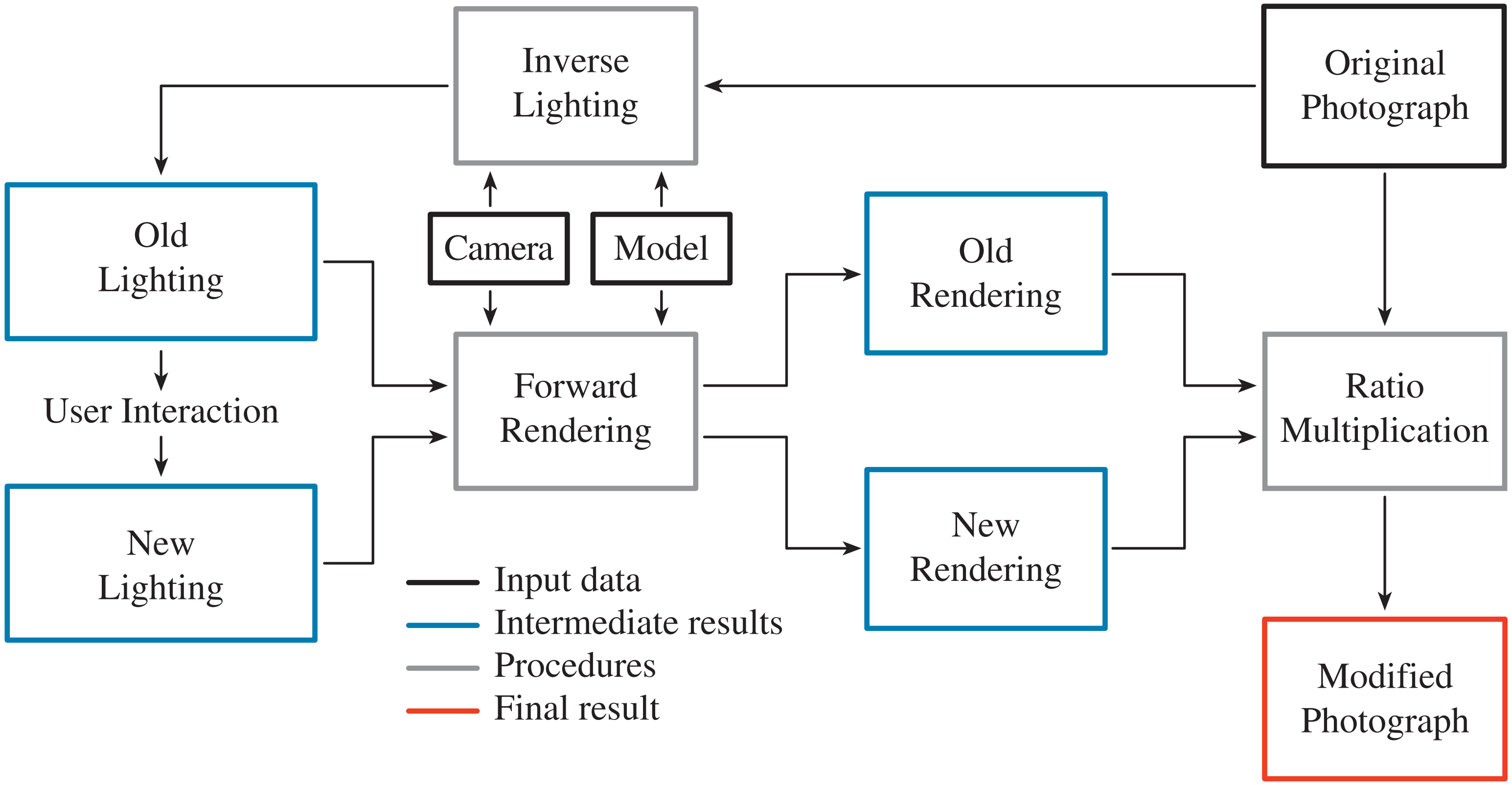


Figure 3.3: The data flow in the re-lighting system.

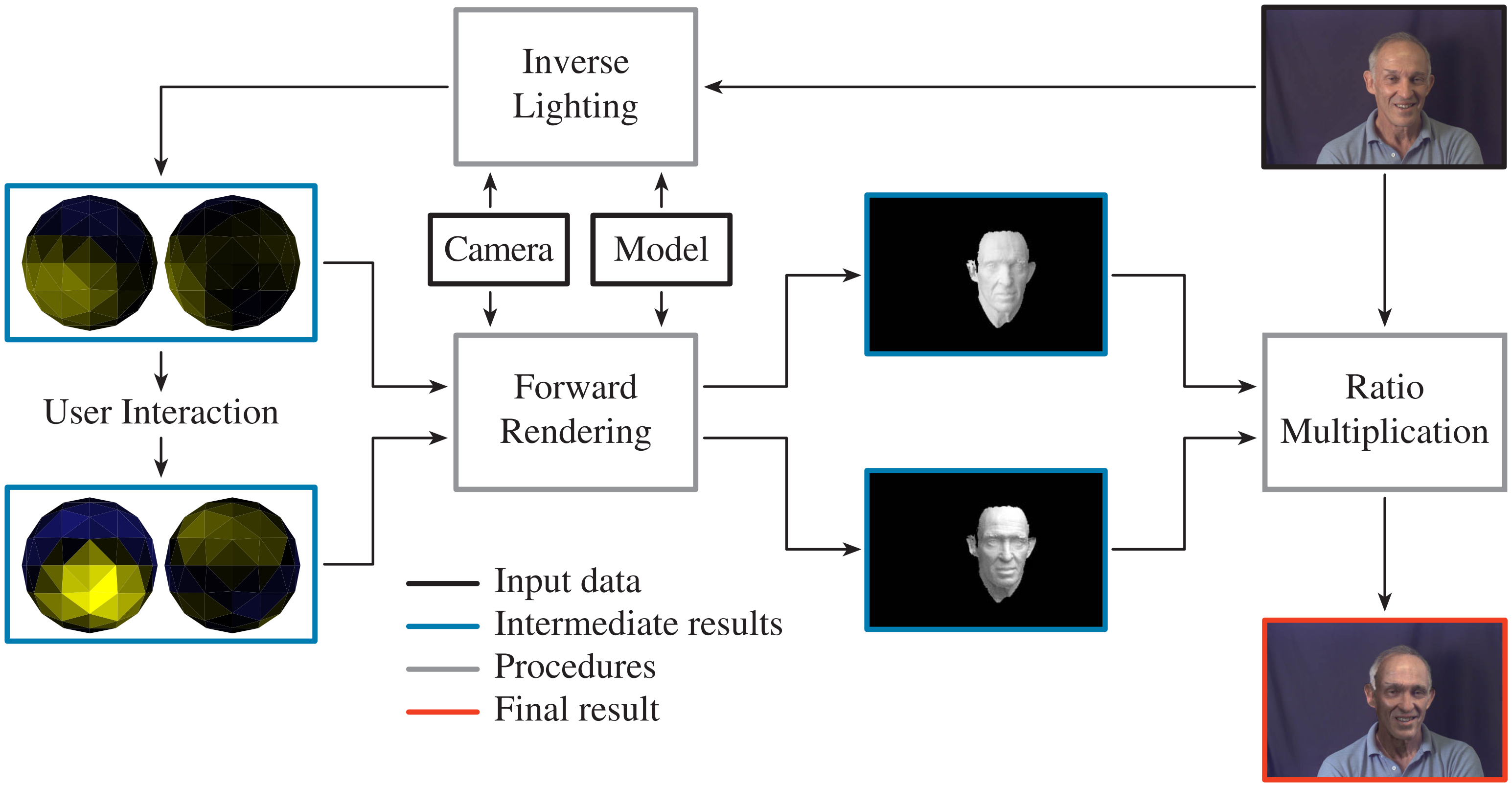


Figure 3.4: The data flow in the re-lighting system.

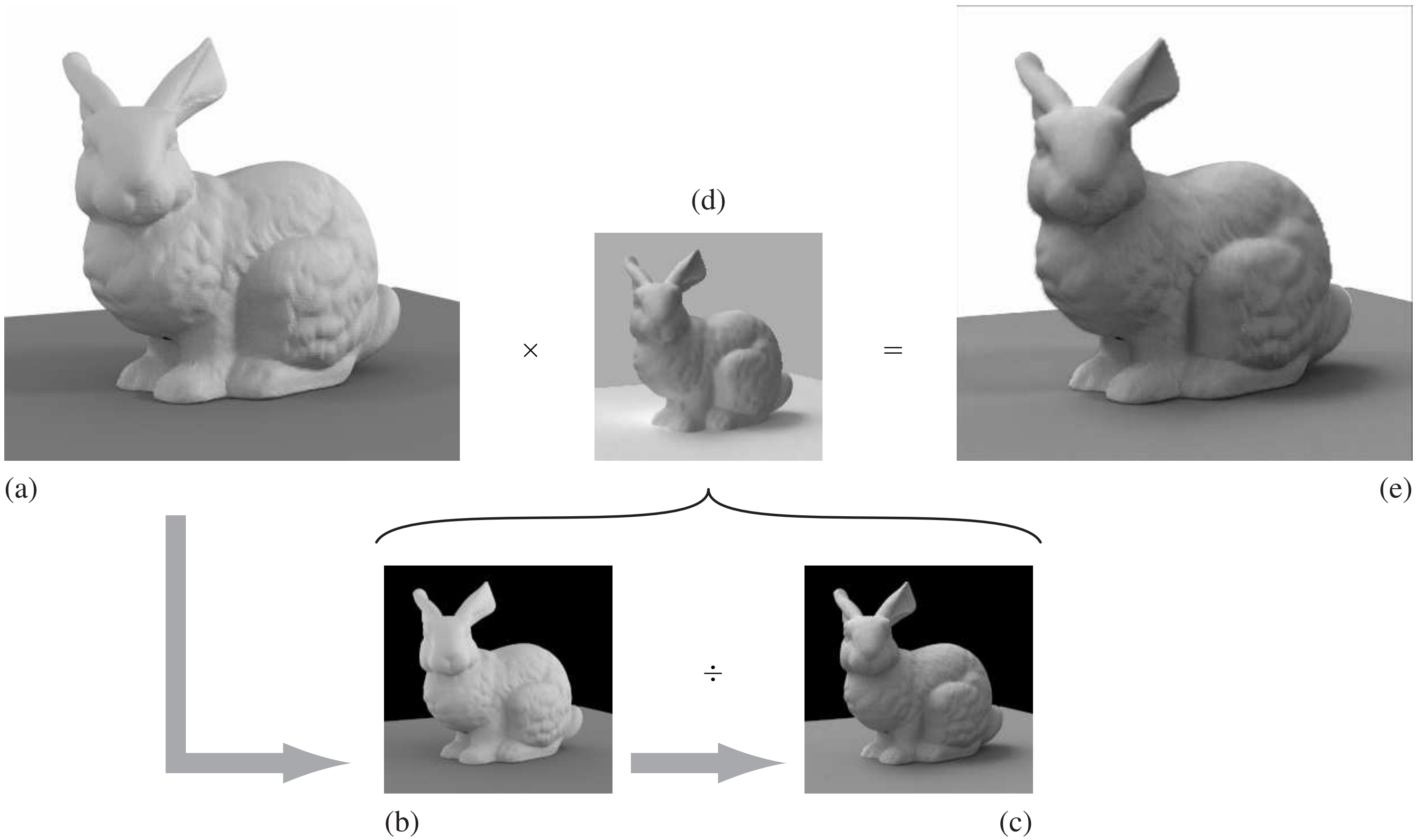


Figure 3.5: The results of testing the re-lighting algorithm on a synthetic photograph. (a) The original "photograph." (b) The 3D model rendered under the lighting configuration computed using inverse lighting; this is the old rendered image. (c) The new rendered image. (d) The ratio of (c) to (b). (e) The modified photograph.

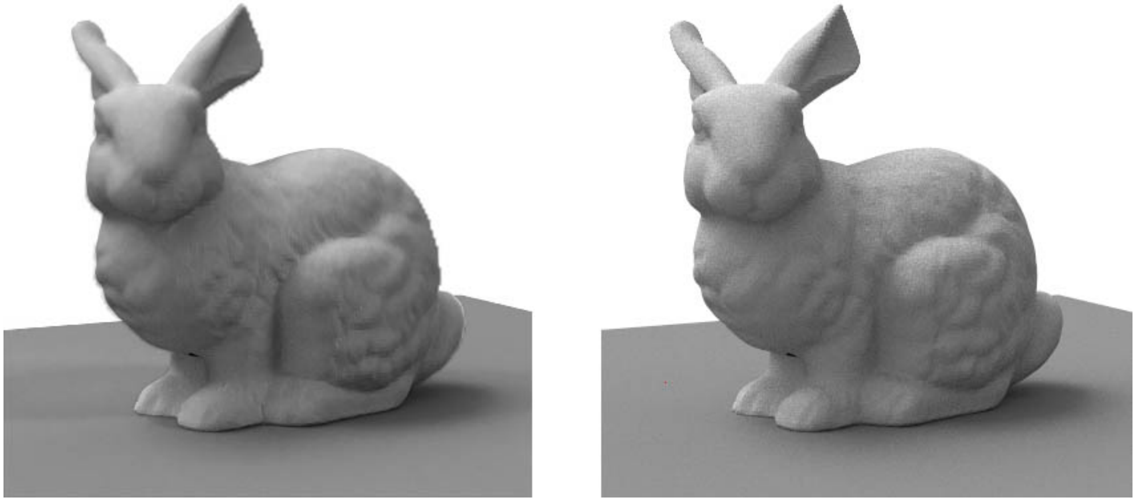


Figure 3.6: The re-lit image from Figure 3.5 (e), left, compared with an image rendered directly from the new lighting, right. The images are very similar, but note that the shadow has not been completely removed and that there are some aliasing artifacts along the edge of the model.

3.7 A Test with a Rigid Object

The second example of re-lighting is a picture of a well-behaved real object, a ceramic planter that we painted with gray primer. We used a Cyberware 3030MS scanner (Appendix E) with the Zipper and VRIP software packages (Section 4.1.2) to get the geometric model, and we used a grey, Lambertian model for the reflectance.

3.7.1 Camera Calibration

Because we are now dealing with real images, it is important to consider how the photograph is related to the 3D scene.

The camera we used for this section is the Photometrics PXL 1300L, which is described in Appendix D. We used the lens's nominal focal length, adjusted for the camera's indicated focus distance using the thin lens approximation, for the image plane distance.

In order to establish the correspondence between the photograph and the surface of the object, we needed to measure the position of the camera, relative to the coordinate system in which the scanner measures the object. We achieved this using a planar sheet of the targets described in Appendix C. We scanned it to establish the positions of the targets in the scanner's coordinate frame, then photographed it at two rotations to get a well-spaced set of points from which to calibrate the camera using the camera calibration algorithm described in Appendix A.

3.7.2 Results

The existing lighting in the photograph came from a large source to the left of the camera. We changed the lighting to make it come from above and to the right side. The original and modified photographs are shown in Figure 3.7. The result is convincing, although misregistration causes some silhouette artifacts.

3.8 Tests on Human Faces

Our final examples deal with the important but challenging task of re-lighting human faces. In one example, we illustrate an application of re-lighting to image compositing, as shown in Figure 3.10. We began with a snapshot of a person under office lighting (a), and we wished to paste in a photograph of a second person, taken under different lighting (b); our goal was for the subjects to appear to be standing next to one another. Simply compositing the images would be unconvincing because of the mismatched lighting (c). Instead, we used inverse lighting to compute the light distribution in the office scene. With that information, we re-lit the second photograph to match the lighting in the first before compositing the images (d). In two other examples, we simply modified the lighting on a face by directly specifying the new lighting configuration.

We used scans from a Cyberware 3030PS scanner (Appendix E) for the geometric model, and we again assumed a grey, diffuse surface reflectance. Of course, this is only a rough approximation of the BRDF of human skin, and a better model could be substituted in the future without changing the algorithm.

There are two problems to be overcome before the Cyberware model of a head can be used successfully. One is that much of the head (the hair, ears, and eyes in particular) cannot be measured accurately by the scanner, and the resulting data must be ignored; the other is that the remaining part of the scan does not correspond exactly to the photograph because of movements made by the subject between the time of the scan and of the photograph. We handled the first problem by filtering the geometric data and the second by image warping.

3.8.1 Filtering out unreliable data

In order to reduce the worst errors in the basis images, we used some heuristics to limit the system to pixels corresponding to geometry that can reasonably be expected to be accurate. For scans of human heads, this especially meant finding and ignoring the hair. We generated a binary mask marking the usable regions of the scan as follows (Figure 3.8):

1. Compute a map of the local surface curvature at each sample point.

2. Filter the map with an iterated median filter (size 3 by 3; 3 iterations).

(a)

(b)

Figure 3.7: Re-lighting a photograph of a rigid object. (a) The original photograph; (b) the re-lit photograph.

3. Threshold to separate smooth (low curvature, reliable) regions from rough (high curvature, unreliable) regions.

4. Find and remove isolated smooth regions smaller than a given area.

5. Find and remove isolated rough regions smaller than a given area.

For the purposes of inverse lighting, we used a small threshold area for the last step, since our purpose was to avoid including any unreliable areas. However, for the purposes of computing the change in lighting (Section 3.5) we used a large threshold, since the edges of the small rough areas are visually objectionable in images.

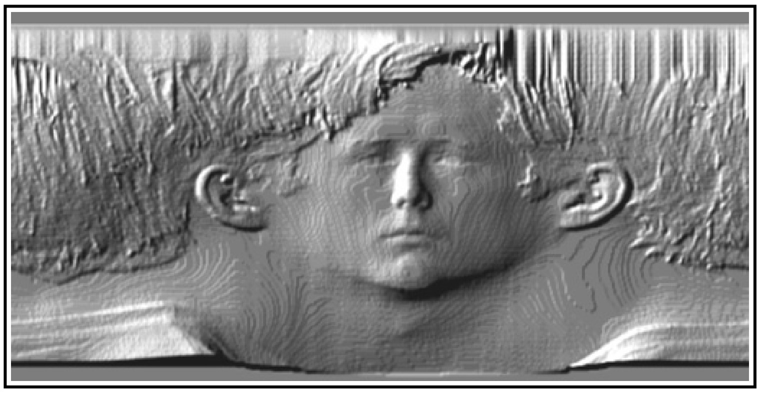
3.8.2 2D image warps to correct registration

In some cases it is not possible to get the renderings into perfect agreement with the photograph; this is particularly true with deformable objects like human faces, since only rigid body motions can be accounted for by moving the camera. In order to avoid the distracting artifacts that are caused near sharp edges (such as the silhouette of an object) when the renderings are out of register with the photographs, we used a feature-based image warping technique by Beier and Neeley [8] to manually distort the renderings to bring them into alignment with the photograph. We then treated this warp as part of the camera model during re-lighting. It might be possible to automate the specification of this warp, especially for situations where the registration errors are very small.

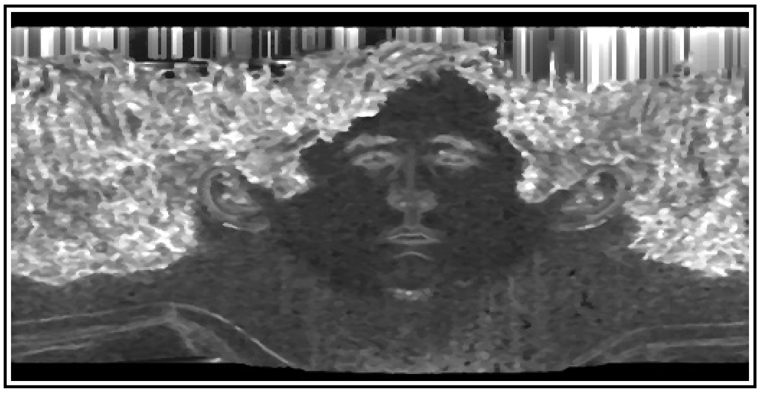
3.8.3 Results

The re-lighting results for the compositing test are shown in Figure 3.9. The illusion in the area of the face is reasonably convincing to the eye, but there are two artifacts that detract from the effect. First, because we are using a Lambertian model for the skin's BRDF, we cannot properly account for any non-Lambertian features in the image. For instance, there is a specular highlight on the subject's right cheek that was caused by the light coming from the left of the image. The highlight persists, though reduced, in the modified photograph, even though the light that made the highlight has been moved elsewhere, and no new highlight appears from the new light source position.

The second, and more obvious, artifact is the stark contrast between the darkened side of the face and the adjacent, unmodified area of hair. The image is modified only in the area that is occupied by the model, which cuts off at the edge of the hair. This leads to a sharp dividing line at the edge of the model in areas where the image is being changed significantly. Not only are these sharp edges very noticeable to the human eye, but by providing bright reference tones they create the appearance that the modified skin areas



(a)



(b)

(c)

(d)

Figure 3.8: Results of filtering range data. (a) The range data from a typical scan; (b) the filtered curvature map computed from that data; (c) the mask used for inverse lighting; (d) the mask used for lighting modification.

(a) (b)

Figure 3.9: Re-lighting a photograph of a human face. (a) The original photograph; (b) the re-lit photograph.

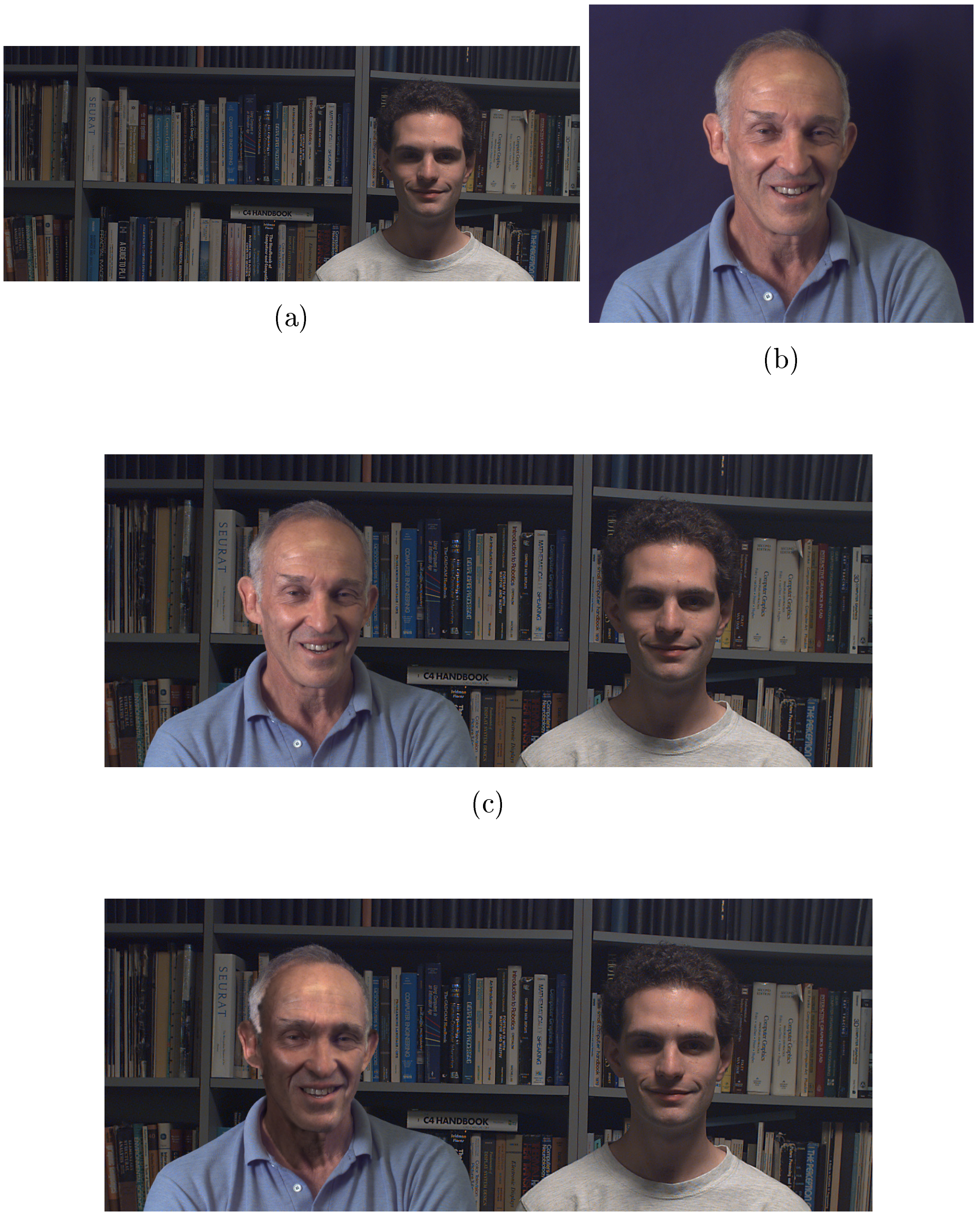
nearby have a lower reflectance than the rest of the face. This destroys the desired illusion that they are lit less brightly.

Two more examples of re-lighting faces are shown in Figures 3.11 and 3.12. The original photographs were taken using a large rectangular area source behind and above the camera, producing a symmetrical, frontal lighting. In the first case, we replaced this source with a large area source to the subject's left, and in the second case we specified light coming from overhead. Again, the results in the area of the faces are convincing, and the subjects' dark hair greatly reduces the artifacts at the edge of the forehead. In Figure 3.11, the subject's evenly lit shirt causes the same type of context problems that we saw in the previous example: note how covering up the bottom part of the image greatly improves the illusion. Also, the problem of specular highlights causes the results to appear subtly unrealistic in some areas.

3.9 Conclusion

In this chapter we have demonstrated the ability to solve an inverse rendering problem to find the lighting in a photograph. The algorithm combines the strengths of physically-based rendering with techniques for solving linear inverse systems. We have also demonstrated the application of the results of inverse lighting to the task of modifying lighting in photographs, both in controlled test cases and in a very challenging case that is important to photography, that of a human face.

The ability of these algorithms to use geometric data to understand and manipulate lighting in three dimensions illustrates the possibilities of processing photographs armed with this kind of additional information. Given the rapidly advancing field of 3D scan-



(a) (b)

(c)

(d)

Figure 3.10: Using re-lighting as an aid to compositing. (a) The source photograph; (b) the destination photograph; (c) the photographs composited directly; (d) the photographs composited with re-lighting.

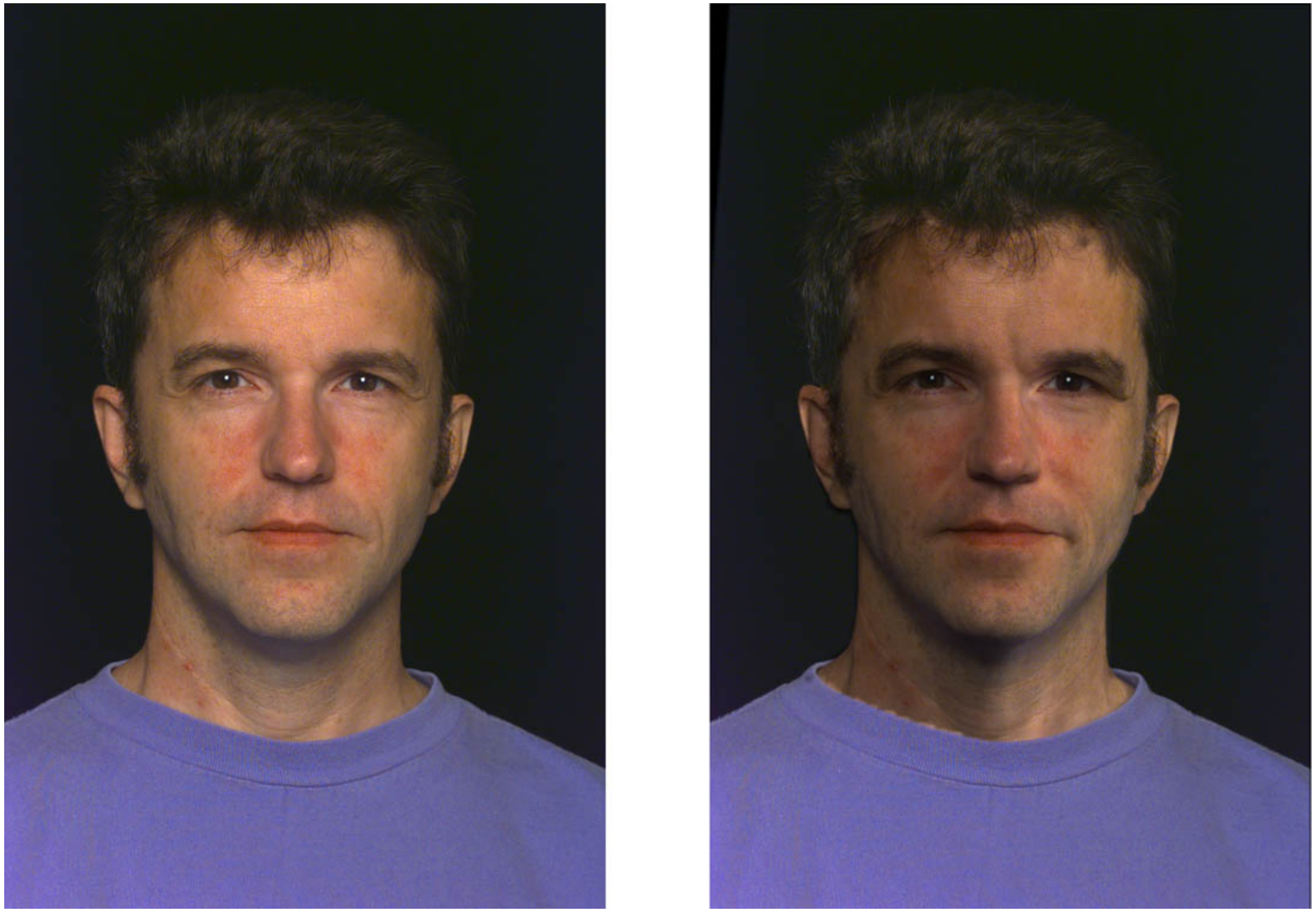


Figure 3.11: The results of re-lighting a face to move the light from the front to the side.

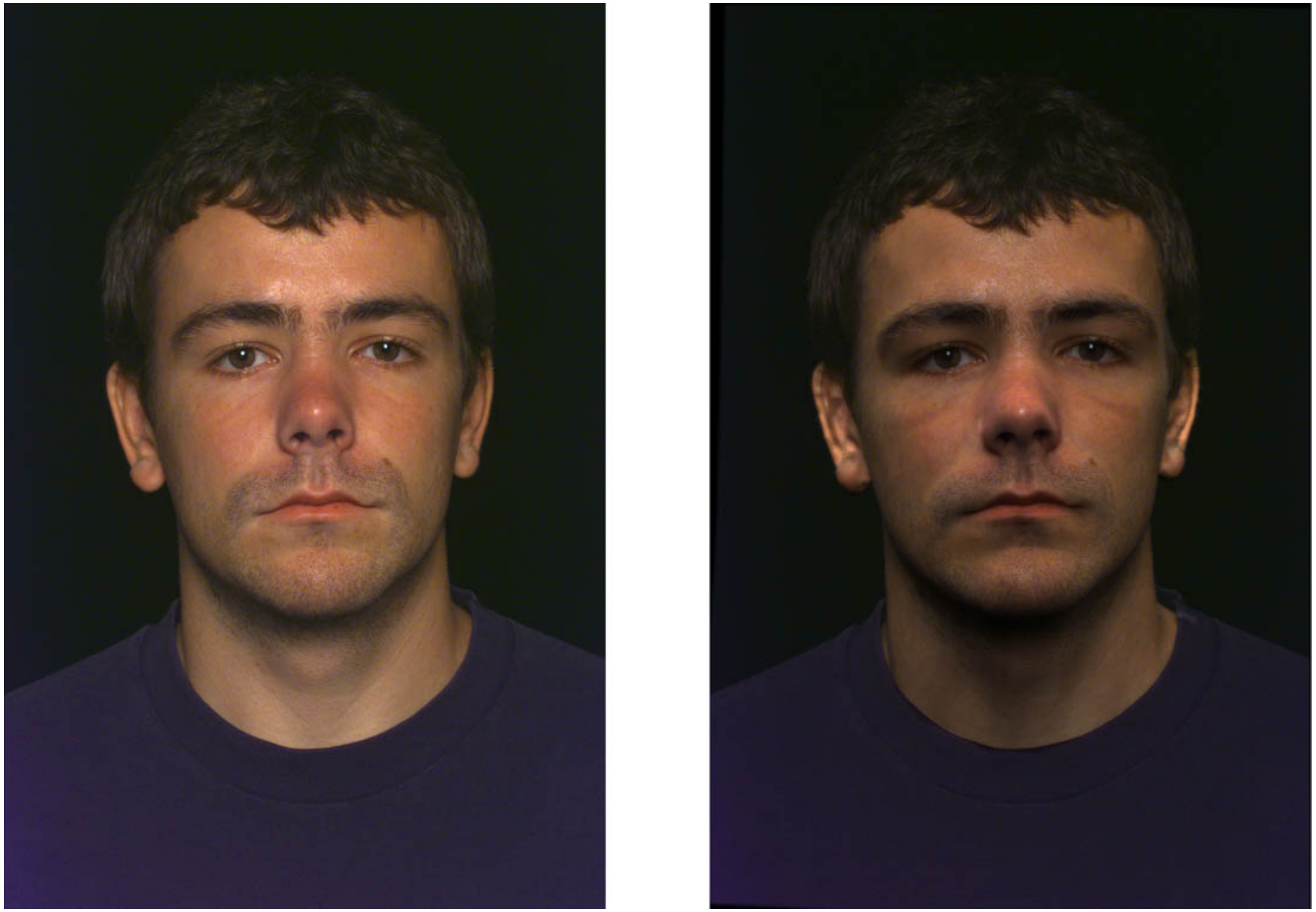


Figure 3.12: The results of re-lighting a face to move the light from the front to above.

ning and range imaging, future digital cameras could incorporate the hardware required to measure 3D geometry in the same way that current cameras incorporate flashes to provide illumination. This would allow for a variety of graphics operations using the geometry itself, but, as this chapter demonstrates, it would also open up new possibilities for processing the high-resolution photograph as well.

3.9.1 Future work

Many models for BRDF have been presented in the computer graphics literature, and the mathematics of our system is capable of handling models more sophisticated than the Lambertian model used in this chapter. Of particular interest with regard to faces is Hanrahan and Kreuger's work on the BRDF of materials like skin [29].

We chose the simplest possible basis for L_b, consisting of piecewise constant functions that cover the sphere uniformly. Other bases might well be used to advantage, both smoother bases and more specialized bases that concentrate the detail in areas of greater importance.

The image warps we use to correct for errors could be fit automatically using an edge-alignment-based technique, especially if the camera is very nearly aligned to start with. It might also be interesting to work with a generic head model that doesn't match the subject very well at all but depends on the image warping to account for the differences.

In cases where the geometric model does not describe the entire object being re-lit, our practice of ignoring all pixels outside the image of the model leads to the introduction of sharp features at the edges of the modeled surface. To make it practical to re-light photographs with incomplete models, we must find ways to make the transition less visually apparent without introducing still more artifacts in the process.

Our image ratio technique assumes that all unpredicted features in the image are due to variations in surface color. Although this is often true, details in an image are also caused by fine geometric detail, and this type of variation is not handled well by the ratio technique.

Chapter 4

Photographic Texture Measurement

A great deal of work in the fields of optical engineering, computer vision, and computer graphics has gone into developing machines and algorithms to precisely measure the geometry of 3D surfaces. With the tools that have resulted, we can scan an object at sub-millimeter resolution and assemble the resulting data into a complete 3D surface model ready to be used for rendering, computer-aided design, or other purposes.

For realistic rendering, however, these models lack an accurate description of the BRDF of the surface, so that physically meaningful results are impossible. Some scanning methods do provide a spatially varying value for the surface color, but this value is normally a measurement of radiance reflected in a particular direction under particular lighting conditions, rather than a measurement of an intrinsic property of the surface. For example, techniques that compute geometry by analyzing full-frame images often use the pixel values from those images as measurements of "surface color" [51, 42, 50]. Some laser triangulation systems, like the Cyberware scanner (Appendix E) used in this work, return colors from a separate camera, using illumination independent from the triangulation laser, while others, such as the system of Rioux et al. [52], use a pseudo-white laser so that color information may be gathered directly from the same signal used for triangulation. In all these cases, the texture map that describes surface color includes both lighting-dependent effects, such as shadows and shading on curved surfaces, and view-dependent effects, such as specular highlights[1] and retroreflection.

These faults may be remedied in part by using very uniform lighting. If scanning could be done in a perfectly uniform radiance field—a difficult undertaking at best—then the

[1]In this chapter, we will adopt a widespread usage and loosen the term "specular reflection" to include the specular-like part of directional diffuse reflection. By this we mean a component of directional diffuse reflection that occurs for configurations close to specular configurations.

texture maps so derived would measure hemispherical-directional reflectance, a significant improvement over the situation with nonuniform lighting. In the case of a convex diffuse surface, this would in fact be an intrinsic value sufficient to describe the surface's reflective properties. Unfortunately, any non-convex object will always shadow itself, leading to inaccuracies even in the presence of uniform illumination.

The problem that must be solved, then, is to start with measurements of radiance reflected in certain directions at various points on the object's surface and end up with an intrinsic description of the surface's reflectance—one that is valid for any viewing direction and any lighting configuration. In terms of the rendering equation presented in Chapters 1 and 2, we need to start with part of f and solve for $\mathbf{K}$. Since we have scanned the object's geometry, $\mathbf{G}$ is known, and if we also know h we have a workable inverse rendering problem.

Solving for $\mathbf{K}$ in full generality by measuring a whole BRDF at every point is too great an undertaking. That would require hundreds of measurements at every surface point, and since each photograph provides at most one view of a point, we would have to take an even larger number of photographs. Because rendering requires full spatial detail for visual richness, we chose to retain high spatial resolution while reducing the amount of data required by representing the BRDF at each point using a BRDF model with a small number of parameters. In fact, in many of the practical examples to follow, we have used the simplest possible, one-parameter BRDF model: a constant, representing a Lambertian surface.

To obtain measurements of reflected radiance, we opted to use still photographs from a digital camera, rather than using the color values from the scanner, because of the greater flexibility, speed, resolution, and dynamic range that a high-quality digital camera can deliver. Thus, our input data consisted of a set of photographs of the object, each from a known camera pose and each with known lighting. The inverse rendering problem was to construct, based on the samples of f provided by the photographs, a representation of $\mathbf{K}$ in terms of the spatially-varying parameters of a BRDF model.

4.1 Prior Work

4.1.1 Texture mapping

Since the early days of computer graphics, *texture maps* have been used to enrich renderings by using images to modulate the color of surfaces [11, 31]. In the context of physically realistic rendering, texture mapping means using one or more images to control the spatial variation of a surface's BRDF.

Images for texture mapping can come from a variety of sources. Many textures come from photographs or artists' illustrations, and in these cases the images are always stored in a sampled representation. Textures can also be generated algorithmically, in which case they can be computed and sampled ahead of time or generated on the fly as they are needed.

Whatever their origin, an important problem in the use of texture maps is defining how the image is mapped onto the surface. Most texture maps, whether procedural or photographic, are inherently flat in that they are computed, drawn, or photographed in a plane, and they must be distorted to be mapped onto any curved surface that is not developable (cannot be formed by rolling a piece of paper). Some exceptions to this rule, textures that are not inherently flat, are procedural textures computed using the proper distance metric for a particular curved surface [62, 66, 23, 40], textures from 3D paint programs [28, 2], and solid textures. The texture maps we describe in this chapter, being measured from the actual surface itself, also fall into this category, and they have no distortion when mapped onto the surface. All we require is a mapping to a flat domain, and any geometric distortion induced by that mapping will be exactly canceled by the mapping back to the surface for rendering.

How a texture map is mapped to the surface is described by a function $\phi : S \subseteq \mathcal{M} \to D \subset \mathbb{R}^2$ that associates some point in the texture map (defined on the domain D) with every point in the area S of the surface that is covered by the texture map. We call ϕ a *texture embedding*. It allows the renderer to look up the appropriate information from the texture map when performing reflection calculations at the surface. There are a number of techniques for defining texture embeddings [31, 10, 9], but the resulting functions can have a variety of problems. Embeddings that produce extreme geometric distortion can force the use of very high resolution texture images to maintain an adequate sampling rate across the entire surface. Few methods can construct functions that are bijective (that is, each surface point maps to a single texture point and vice versa) for arbitrary surfaces. Also, discontinuities at the edges of texture maps lead to troublesome boundary conditions and possible discontinuities of texture on the surface.

All of the properties just mentioned—low distortion, bijectivity, and continuity—are important to our algorithm for constructing textures on scanned surface models. The mathematics of smooth manifolds guarantees that all these properties can be achieved locally, in a neighborhood of any point. This leads us to use an approach to representing functions over manifolds that is often called an *atlas* of *charts*. This method is described in Section 4.2.

4.1.2 3D scanning and reflectance modeling

Our work builds upon techniques for generating accurate, detailed 3D models of real objects, particularly Turk and Levoy's Zipper [63], a system for aligning triangle meshes in space and stitching them into surfaces, and Curless and Levoy's VRIP [15], a program that uses a volumetric intermediate representation to merge several range images into a single surface. It also builds on existing surface parameterization techniques [18].

Debevec, Taylor, and Malik, in their work on rendering architectural models [16], also used photographs to construct texture maps for models of real objects. However, the geometric models they used were much simpler than the scanned models used in this chapter. Also, they did not attempt to measure the surface's intrinsic reflectance, instead modeling the reflected radiance under the prevailing illumination, although they did use view-dependent texture mapping to allow for directional variation in the reflected radiance.

The field of computer vision has also dealt with the relationship between reflectance and radiance recorded in photographs. Nayar, Ikeuchi, and Kanade [44] have addressed the problem of measuring reflectance in the presence of interreflections, in the context of shape from shading via photometric stereo. Under the assumption of diffuse surfaces, they infer both geometry and reflectance using several photographs from one camera position with different, known, light sources.

As we do in this work, Sato and Ikeuchi [53] have determined spatially varying surface properties from images of a scanned object. Their experimental setup is similar to the one used in this chapter, but they concentrate on fitting a dichromatic (diffuse plus specular) reflection model to determine a single material for each of a few regions. Similarly, Baribeau et al. [7] work with registered range and color images, estimating parameters of a dichromatic model for one or more objects with uniform material properties. In contrast, we find high-resolution texture maps without dividing the surfaces into regions according to BRDF. In later work [54] Sato, Wheeler, and Ikeuchi used their system to estimate spatially varying dichromatic reflectance, addressing a similar problem to the one in this chapter. However, their approach to handling the fundamental difficulties (described in Section 4.3.2) associated with measuring spatially-varying specular reflection is rather *ad hoc*, and the test case they present is a primarily cylindrical object, which greatly simplifies the problem by producing a linear specular highlight that sweeps across the whole surface. Also, they rely on separating the specular and diffuse components in color space, which inherently depends on having a saturated diffuse color.

Ofek et al. [47] presented an algorithm to construct texture maps using measurements from a sequence of photographs. Their algorithm's main strength is its ability to account for the different sampling rates that result from very different views of a surface. They use robust statistics to ignore specular reflections, but they do not attempt to model the surface's reflectance; rather, their texture maps record the diffuse component of the radiance reflected from the surface under the lighting conditions at the time the photographs were taken. Their technique works in principle for arbitrary surfaces, but they only demonstrate it on planes and cylinders, and they do not discuss the difficulties of handling more general geometry.

Some existing range scanners (including the scanner used in this work) do provide texture maps with the surface models they generate, but the values in the texture maps record the radiance reflected from the surface, rather than the reflectance information required for rendering [52].

4.2 Texture Representation

The output of our algorithm is **K**, represented by a function $\mathcal{M} \to \mathbb{R}^p$, where $\mathcal{M}$ is the manifold representing the object's surface and p is the number of parameters to the BRDF model. Representing a function on a geometrically complex domain with arbitrary topology requires some care, particularly if we hope to avoid highly irregular sampling density. As described in the previous section, we used an atlas of overlapping charts that together cover the surface. We define separate functions on each of a collection of overlapping domains, which we call *texture patches*, that together cover the entire surface, but we require the functions to agree where their domains overlap. This idea is likened to an atlas of the world that is composed of many maps, or charts: each chart covers part of the Earth's surface continuously and with low distortion, and together the charts map the entire globe. Any given point can be found on one or more charts, and since the charts agree where they overlap we may consult any of those charts with the same result.

To describe **K** by this method, we must define n *texture patches*, $S_i \subset \mathcal{M}$, each homeomorphic to the open disc, whose union covers all of $\mathcal{M}$. In addition, we need n homeomorphisms, called *texture embeddings*, $\phi_i : S_i \to D$, where D is the open unit disc in $\mathbb{R}^2$. We would also like ϕ_i to induce as little metric distortion as possible (that is, angles and relative lengths should be preserved as well as possible). To complete the representation we define *texture maps* $m_i : D \to \mathbb{R}^p$. Then for a surface point $\mathbf{x} \in S_i$ the corresponding BRDF parameters are $m_i \circ \phi_i(\mathbf{x})$. Figure 4.1 illustrates S_i, ϕ_i, and m_i. Since $\mathbf{x}$ may belong to more than one region, the maps must agree where they overlap: if $\mathbf{x} \in S_i$ and $\mathbf{x} \in S_j$ then $m_i \circ \phi_i(\mathbf{x}) = m_j \circ \phi_j(\mathbf{x})$. In practice, since we use a sampled representation of m_i, this equality will not be exact; rather the two sides should agree up to the precision of our representation.

To compute these three parts of the output—the texture patches, the texture embeddings, and the texture maps—we first compute the S_i's by dividing the surface into disjoint patches and adding an extra layer of triangles to each (Figure 4.2). Then we compute the ϕ_i's using a linear least squares computation to approximate a *harmonic map* to the unit disc (Figure 4.3). The m_i's are the output of the reflectance estimation process, which is the topic of the next section.

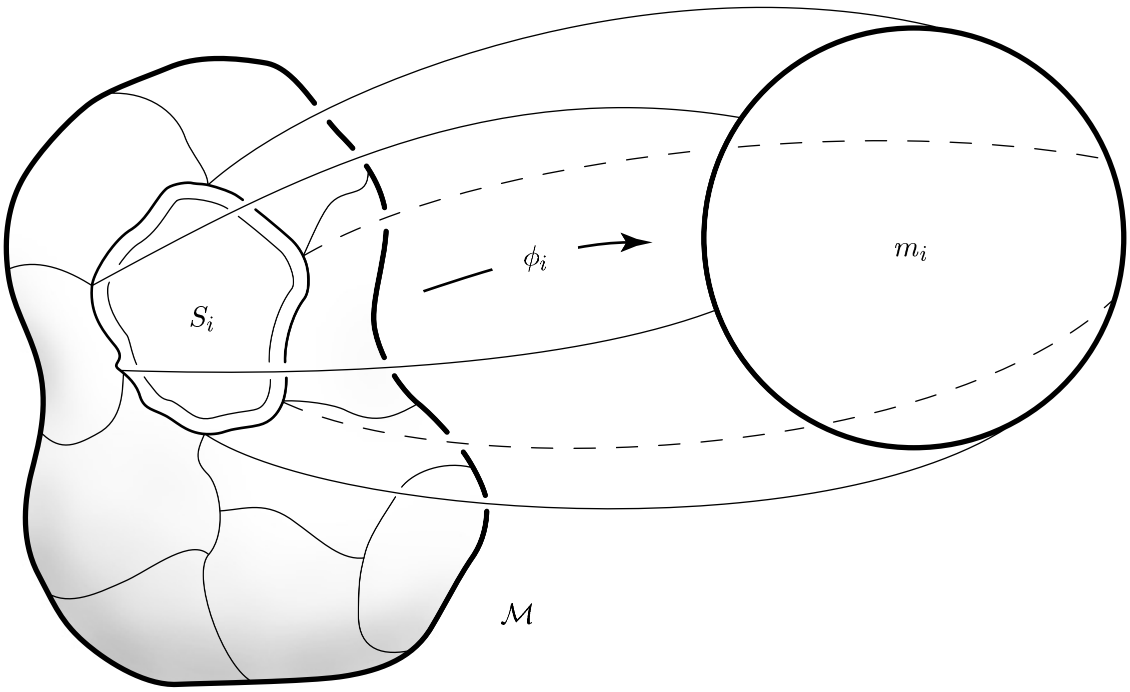


Figure 4.1: How texture patches collectively cover the object. Each of the overlapping texture patches S_i has an associated texture embedding ϕ_i that connects it with the texture map m_i.

We start with a triangle mesh M that represents the topology and geometry of $\mathcal{M}$. The S_i's are computed by first constructing Voronoi-like regions [18] on the surface, starting from a Poisson-disc-like distribution of seed triangles. The seed triangles are chosen by the following algorithm.

$i \leftarrow 0$
Unmark all triangles
WHILE $\exists$ unmarked triangles
    Choose a random unmarked triangle t_i
    Mark all triangles within a distance r of t_i
    $i \leftarrow i + 1$

The triangles $t_1, \ldots, t_n$, which are all separated by a distance of at least r, are the seed triangles. The parameter r is chosen by the user. The Voronoi-like region S_i' then contains all triangles that are closer to t_i than to any other seed triangle (triangles equidistant from two or more seed triangles may be assigned arbitrarily)[2]. This partition of M can be computed in linear time using breadth-first search [14].

[2]Distances between two triangles are measured by the minimum number of edges that must be crossed to get from one triangle to the other (that is, path length in the dual graph to M).

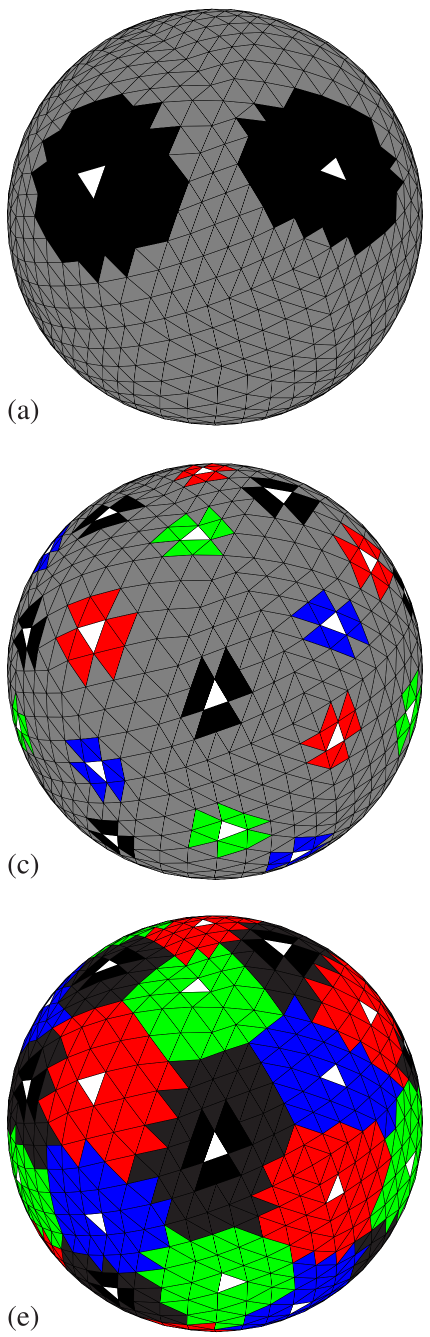


Figure 4.2: Building texture patches on a sphere. (a–b) The seed triangles are chosen in a Poisson-disc-like distribution. Each time a triangle is chosen, a region surrounding it is marked. Each new seed triangle is chosen from the unmarked portion of the surface. This phase completes when the entire surface is marked. (c–e) Voronoi-like regions are formed around each seed triangle by performing a parallel breadth-first search across the surface. Once the maps have grown to cover the whole surface (e), this phase is complete. (f) To each texture patch a layer of triangles is added, so that the final patches overlap.

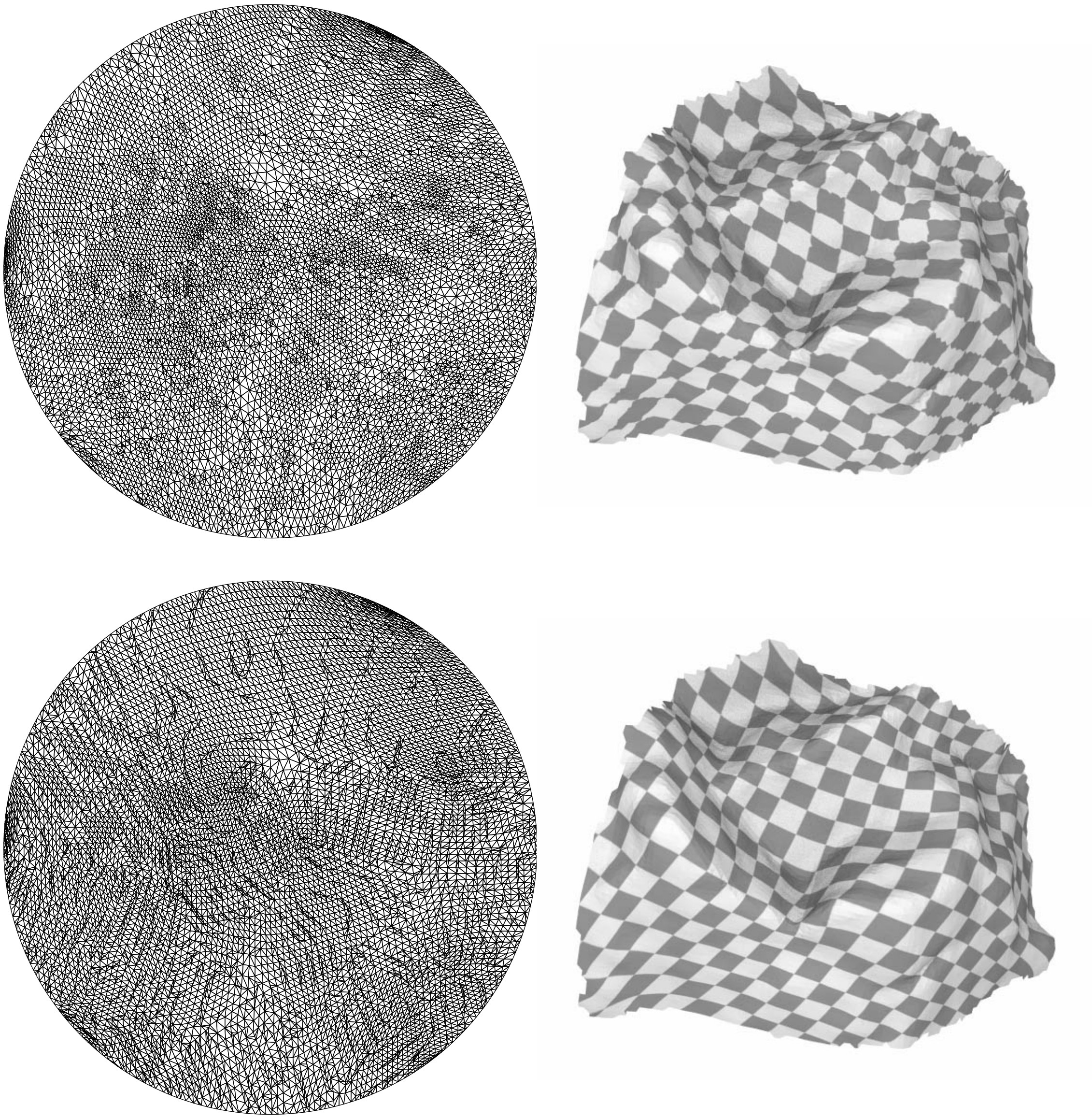


Figure 4.3: Constructing the harmonic map from a texture patch to the unit circle. The first step is to minimize the squared sum of the edge lengths in the texture domain (top row), and the second step is to use the harmonic edge weights to refine the solution (bottom row). The left column shows how the triangles in the texture patch are mapped to the unit circle, and the right column shows a regular checkerboard in texture space mapped onto the surface. Both algorithms produce low large-scale distortion, but the jagged edges in the upper row indicate high local distortion in the unweighted solution. This is expected, since the unweighted solution does not take into account the shapes of the triangles in the 3D model.

To obtain the overlapping patches S_i, we add to each S'_i all triangles that share a vertex with a triangle in S'_i, then add any triangles that are surrounded. Specifically, the procedure is: mark all vertices of triangles in S'_i; mark all vertices of triangles with any marked vertices; find all triangles with all vertices marked and put them in S_i. Note that r must be set low enough so that all patches turn out homeomorphic to the disc; if it is set high enough that a patch can grow to meet itself, this condition may not be satisfied.

Once the S_i's have been chosen, we define ϕ_i for each S_i as a piecewise linear function. For every vertex $\mathbf{x}$ in S_i, we explicitly compute $\phi_i(\mathbf{x})$, known as the *texture coordinates* of $\mathbf{x}$. From these coordinates, ϕ_i is defined for all non-vertex points by linear interpolation. To compute the texture coordinates at the vertices, we first fix the texture coordinates of the vertices on the boundary of S_i at points spaced around the unit circle, with the arc length between points proportional to the length of the edge in M between the corresponding vertices. With these boundary conditions, we then solve a linear system described by Eck et al. [18] to find an approximation to the unique *harmonic map* from S_i to the unit disc, which minimizes *metric dispersion*, a measure of local distortion of shape. The reader is referred to the paper of Eck et al. for details on this system. We solve the large, sparse linear least squares system by using Gauss-Seidel iteration on the normal equations.[3] To provide a starting point for this process, we first run the same iteration with uniform edge weights (thus minimizing the sum of squares of the edge lengths[4]). Figure 4.3 illustrates the results of this system, and demonstrates why it is important to use the edge weights that lead to an approximate harmonic map rather than using a simpler algorithm such as the unweighted system we use to find our starting point.

4.3 Estimating Reflectance

With a surface model, a parameterization of the surface, and a series of photographs, we are ready to compute the reflectance of the surface. Our algorithm first gathers all the observations of radiance reflected from a particular surface point by sampling all the photographs in which that surface point is visible and illuminated. From those measurements, and from the known geometry of the surface, camera, and light source, the incident and exitant directions and BRDF value are computed for each observation. These samples of the surface's BRDF at that point are then used to estimate parameters of a BRDF model. In practice, this means averaging the values, with weights that depend on the incident and exitant directions, to obtain an estimate of the Lambertian component of surface reflection.

[3]The usual cautions about the instability of the normal equations do not apply to this large-residual problem.

[4]This is the configuration that a network of springs with zero rest length would take.

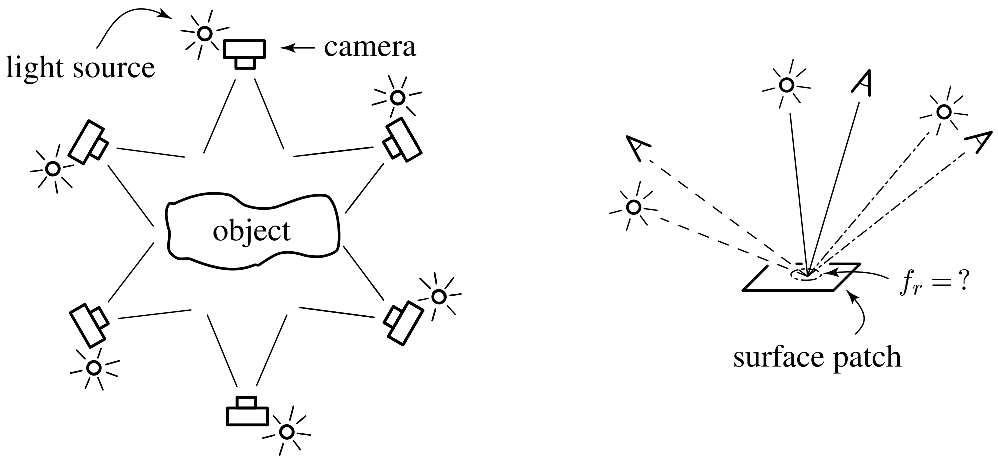


Figure 4.4: At left, the object is photographed from a number of views with point source illumination. At right, the problem of estimating reflectance at a surface point.

For reasons discussed in later sections, the specular part of the BRDF must be handled later by combining measurements from different surface points.

The actual processing of a particular sample point $\mathbf{p}$ in a texture map m_i proceeds as follows. We first compute $\mathbf{x} = \phi^{-1}(\mathbf{p})$, the corresponding point in 3-space, and determine by ray casting which camera positions and light source positions are visible to $\mathbf{x}$. For each photograph for which both are visible, we sample the image at the appropriate location to find the reflected radiance, L_e, and we compute the irradiance I from the distance to the light source. In the absence of interreflections (which we ignore), L_e/I is a measurement of $f_r(\mathbf{x}, \omega_i, \omega_e)$, where ω_i is the direction to the light source position and ω_e is the direction to the camera position. To refer to these measurements in the equations that follow, we will let $\omega_i(\mathbf{p})_k$ and $\omega_e(\mathbf{p})_k$ represent the incident and exitant directions for the k^{th} measurement at the sample point $\mathbf{p}$ (we will use this notation again in Section 4.7). Together the configuration $(\omega_i(\mathbf{p})_k, \omega_e(\mathbf{p})_k)$ is $x(\mathbf{p})_k$, and the corresponding measured BRDF value is $y(\mathbf{p})_k$. Since the number of measurements is different for each point, we must consider it a function of position as well; we will let $n(\mathbf{p})$ be the number of measurements at $\mathbf{p}$. It should be noted that $\mathbf{p}$ is a discrete variable that takes on values only on a regular grid in each texture map's domain.

The task of reflectance estimation, then, is to choose parameters to fit the BRDF to the several measurements of $f_r(\mathbf{x}, \cdot, \cdot)$ (Figure 4.4). To conform to the usual notation of least-squares fitting, we will denote the texture map $m_i : D \to \mathbb{R}^p$ by the symbol a. Thus $a(\mathbf{p})$ is the vector of p parameters that defines the BRDF at texture point $\mathbf{p} \in D$. Let $f_m(a(\mathbf{p}); (\omega_i, \omega_e))$ be the value of the BRDF model for the parameters $a(\mathbf{p})$ at the configuration (ω_i, ω_e). The goal of BRDF fitting is, for each $\mathbf{p}$, to set $a(\mathbf{p})$ so that

$f_m(a(\mathbf{p}); x(\mathbf{p})_k)$ agrees with $y(\mathbf{p})_k$ for $k = 1, \ldots, n(\mathbf{p})$. The least squares approach to this problem is to solve the following problem for each $\mathbf{p}$:

$$\min_{a(\mathbf{p})} \sum_{k=1}^{n(\mathbf{p})} \lambda(\mathbf{p})_k^2 (f_m(a(\mathbf{p}); x(\mathbf{p})_k) - y(\mathbf{p})_k)^2 \tag{4.1}$$

where $\lambda(\mathbf{p})$ is a vector of $n(\mathbf{p})$ weights that control the relative importance of fitting the different measurements. In the case of a Lambertian model, this means fitting a constant function, or, in other words, averaging the estimates. We will assume a Lambertian model for now and revisit the issue of more sophisticated models in Section 4.3.2.

4.3.1 Estimating with a Lambertian BRDF model

When we average the BRDF estimates from the various images of a point, we should weight each measurement according to its reliability. This reliability depends on the geometric configuration of the measurement in three ways:

1. Samples with near-normal illumination are more reliable than samples with near-grazing illumination, because the signal measured by the camera is proportional to $\cos\theta_i$. This gives samples with low θ_i better signal-to-noise ratio than samples with high θ_i.

2. Samples from views nearly normal to the surface are more reliable than samples from near-grazing views, because the density (on the surface) of samples from a particular image is proportional to $\cos\theta_e$. To prevent the blurry texture that results from projecting an image at an oblique angle from overpowering the sharp texture from a near-normal view, the weight must depend on θ_e.

3. Samples that include significant contribution from specular reflection are less reliable than those that do not. This is because we are estimating only the diffuse component of the surface's BRDF. Most pixels in most views of an object that exhibits both specular and diffuse reflection will give no evidence of specularity, so the large majority of measurements can be used safely under a Lambertian assumption. However, those pixels that do show specularity must be down-weighted to avoid contaminating the diffuse component with pasted-on highlights.

We accounted for each of these influences on the reliability of samples by introducing a factor into the weight used to average the samples. The three weighting factors, α_e (for exitant direction), α_i (for incident direction), and α_s (for specular reflection) are given in Table 4.1. The complete algorithm is given as pseudocode in Figure 4.5. In this code, $\alpha(\omega_i, \omega_e) = \alpha_i \cdot \alpha_e \cdot \alpha_s$. Note that the quantities θ_i, θ_e, and β in the expressions for the weights are implicit functions of ω_i and ω_e.

Table 4.1: Geometric considerations affecting the reliability of estimates of diffuse reflection and the weighting factors used to account for them. The parameters k, c, p, β_{max}, and q are set by the user. The expression $x \uparrow y$ is the maximum of x and y. Similarly, $\downarrow$ means minimum. The symbol β denotes the angle between the exitant direction and the direction of specular reflection from the incident direction.

	Weighting factor
Angle of incidence	$\alpha_i = (\cos\theta_i)^k$
Angle of exitance	$\alpha_e = [0 \uparrow (\cos\theta_e - c)]^p$
Nearness to specular direction	$\alpha_s = \sin[\frac{\pi}{2}(1 \downarrow \beta/\beta_{max})]^q$

FOREACH texture patch i
    FOREACH pixel position $\mathbf{p}_j \in D_i$
        $\mathbf{x} \leftarrow \phi_i^{-1}(\mathbf{p}_j)$
        FOREACH camera k
            IF $\mathbf{x}$ is not visible to camera k CONTINUE
            $\mathbf{y} \leftarrow$ projection of $\mathbf{x}$ into camera k's image plane
            $\omega_{i,k} \leftarrow$ direction from $\mathbf{x}$ toward light source k
            $I_k \leftarrow$ irradiance at $\mathbf{x}$ due to light source k
            IF $I_k = 0$ CONTINUE
            $L_{e,k} \leftarrow$ radiance value at $\mathbf{y}$ in photograph k
            $\omega_{e,k} \leftarrow$ direction from $\mathbf{x}$ toward $\mathbf{y}$
        $\alpha_t \leftarrow \sum_k \alpha(\omega_{i,k}, \omega_{e,k})$
        IF $\alpha_t = 0$ CONTINUE
        $m_i(p_j) \leftarrow (\pi/\alpha_t) \sum_k [\alpha(\omega_{i,k}, \omega_{e,k}) L_{e,k}/I_k]$

Figure 4.5: Pseudocode for the texture map construction algorithm.

4.3.2 Using non-Lambertian models

To accurately capture the reflective behavior of most surfaces, a Lambertian BRDF is insufficient. We would prefer to use a BRDF model that can match other important features, most notably specular reflection. Unfortunately, we cannot simply fit a more sophisticated model separately at each point by the same method we used for the Lambertian model.

Some rough calculations based on simple assumptions will show why this is true. Let us take as our example a hypothetical BRDF model with three parameters: diffuse reflectance, R, magnitude of specular lobe, k_s, and width of specular lobe, σ. Assume we use m random camera/light source pairs, so that with minimal occlusion each point will be visible to about half the cameras, and illuminated by about half the light source positions, leading to $\frac{n}{4}$ BRDF samples for a typical surface point. Most of these samples will contain essentially no information that can be used to determine k_s or σ, since these parameters affect only the specular highlights, which occupy a small part of each image. To have any hope of solving for the full BRDF model requires at least three distinct measurements, two of which must contain significant contributions from specularity. Such a set of measurements is not a sufficient condition for success, but it is surely a necessary one.

These requirements on the sample configurations have important implications for the number of data that must be collected. Even for a relatively low-gloss surface that exhibits significant specularity up to 10° from the specular direction, the set of configurations where specularity can be observed is less than 2.5% of the overall BRDF parameter space[5]. If the camera and light positions are randomly and uniformly distributed (and therefore ω_i and ω_e are uniformly distributed[6]), this means that even for the average point to have sufficient data would require 80 BRDF samples per point, or 320 photographs. An elementary probability calculation finds that to expect at least 90% of the points to be solvable requires more than 600 photographs, and to increase that figure to 99% requires more than 1000. For BRDFs with narrower specular lobes the problem becomes much worse, and non-uniform distributions of sample configurations can leave holes even with very large numbers of samples. For non-convex surfaces, occlusion will reduce the number of samples per surface point for the same number of photographs; in fact, there may be areas where no specular reflections can ever be observed.

Clearly, we need a technique for making BRDF estimates at points with too few observations to be solved independently. One approach is to make use of data from surrounding areas, and a way to achieve this is outlined in Section 4.7.

If we take the idea of using data from surrounding areas to its logical extreme, we can use data from the entire surface by assuming that some parameters of the BRDF are spatially

[5] in the measure $\sigma \times \sigma$.

[6] also with respect to the measure $\sigma \times \sigma$.

constant while others vary. If we let the diffuse parameters vary while using a single value for each specular parameter, we will have sufficient data to estimate all the parameters (in fact, the specular reflection is then being measured in much the same way as the BRDFs of Chapter 5). For a material with a homogeneous surface layer over a substrate with spatially varying, Lambertian reflectance, this assumption is appropriate, and it allows us to use all the data about specular reflection gathered from the entire surface to solve for the single set of specular parameters describing that reflection, while still determining the spatially varying diffuse reflectance. In fact, we can use the Lambertian estimation algorithm of the previous section to find the (spatially varying) diffuse component, then fit a BRDF model to the residuals of that fit, which represent the specular reflection.

4.4 A Synthetic Example

To demonstrate the principle of our texture map construction technique, and to verify the correct operation of our software, it is helpful to consider an example in which the solution is known. Figures 4.6, 4.7, and 4.8 show such an example. The photographs used as input to the program, shown in Figure 4.6, were computed by a renderer, using a sphere for the model, a world map as the texture, and 20 randomly chosen camera and light source positions. The texture maps were then constructed using a tessellated sphere with 3.3×10^4 triangles. The resulting maps are shown in Figure 4.7, and the complete model is shown rendered in Figure 4.8. In the maps, some regions can be seen to have artifacts associated with the sphere's tessellation—these are areas where the surface was illuminated and/or viewed only very obliquely. For this example the parameters were set to allow texture to be generated in these areas. Under normal use they would be marked as having too few suitable views, and we would go back to take more photographs of the poorly-observed areas.

4.5 Measurement Setup

We used the Kodak DCS 420 digital still camera (Appendix D) to photograph objects sitting on the Cyberware scanner's turntable (Appendix E), with the Nikon SB-16 flash unit providing near-point-source illumination (Figure 4.9). In order for a set of photographs to be of use in determining texture maps, the camera pose relative to the object must be known accurately, as must the light source position. We established the latter by attaching the light source rigidly to the camera—the position of the source could then be determined from its fixed location relative to the camera. We found the positions of the camera relative to the object by measuring the positions of both the camera and the object in the coordinate

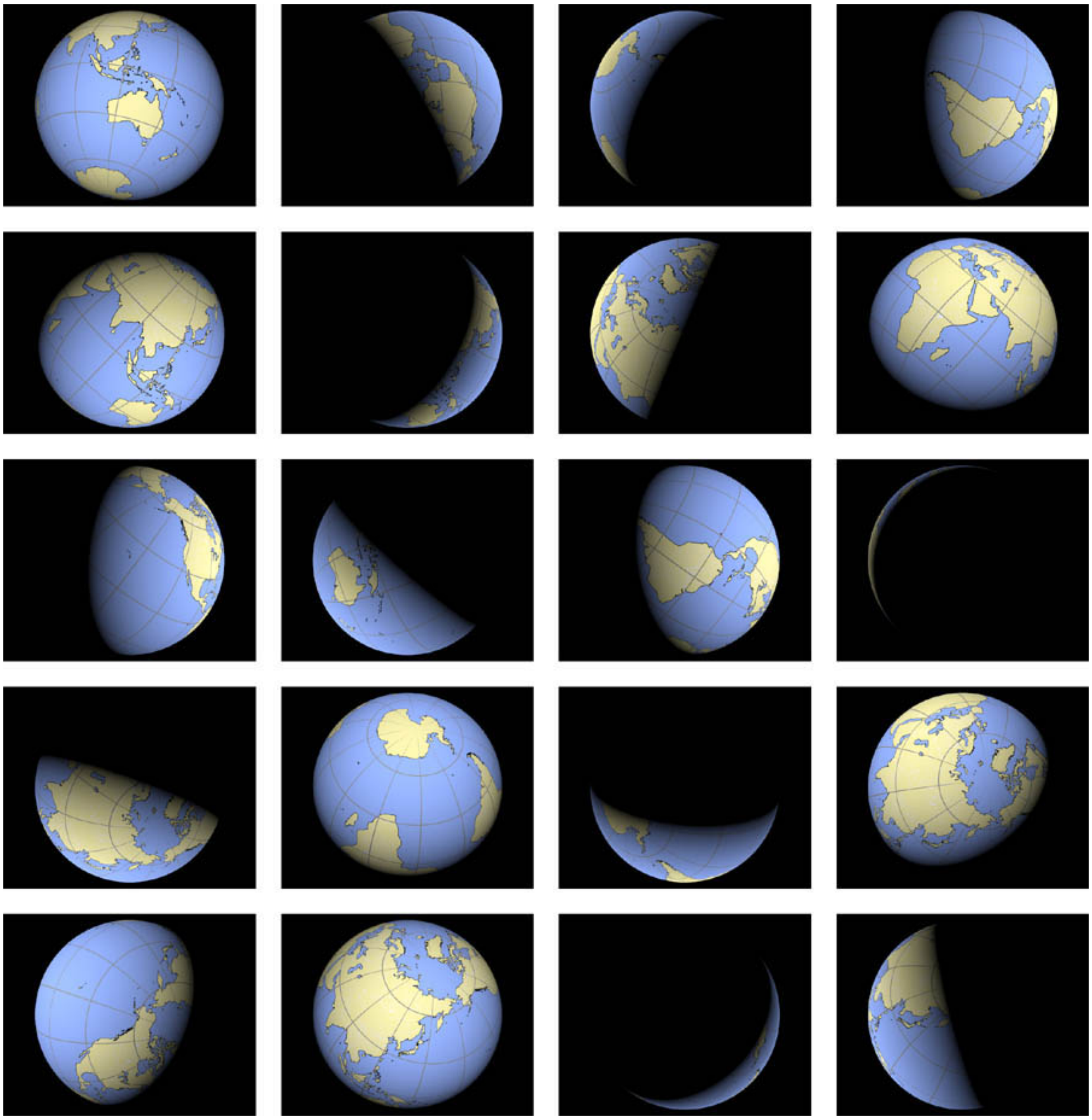


Figure 4.6: The synthetic photographs used as input to test and illustrate the texture mapping algorithm.

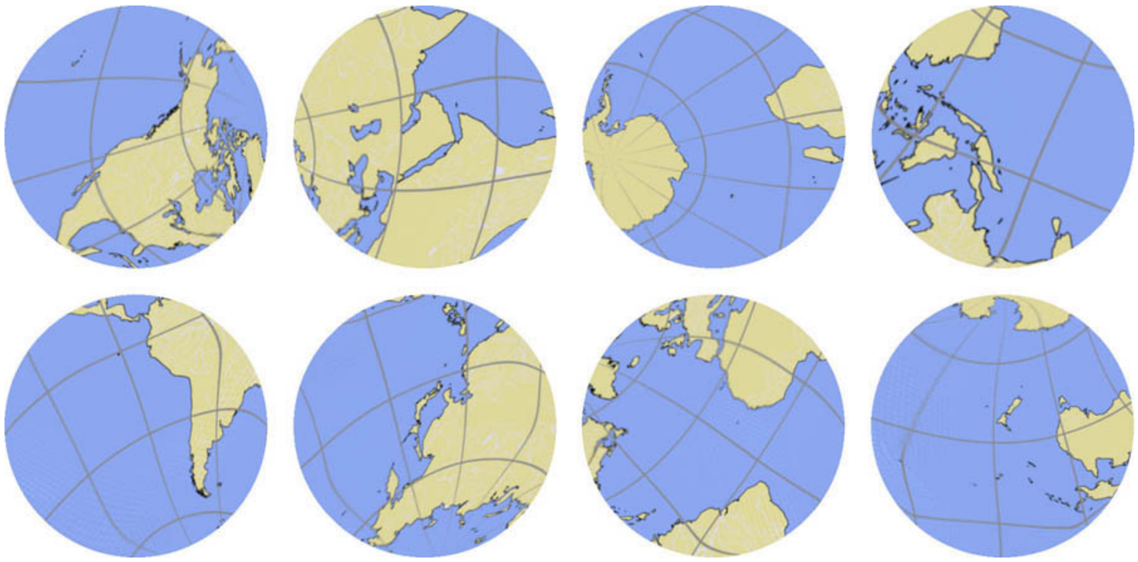


Figure 4.7: The texture maps computed from the images in Figure 4.6.

Figure 4.8: A rendering of the sphere with the computed textures mapped onto its surface.

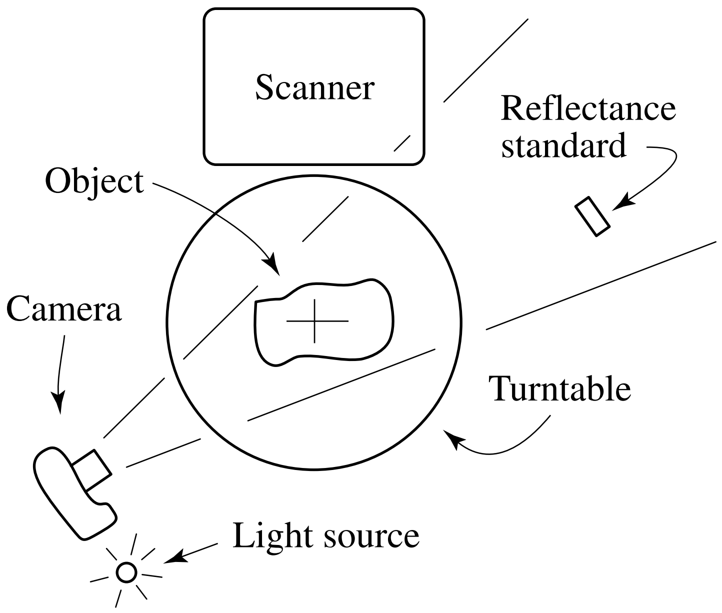


Figure 4.9: The setup used for photographing the objects.

system of the scanner. We located the camera by scanning and photographing a set of targets and using the resulting correspondences between image points and 3D points to determine the camera pose (Appendix A). The object's position was established by taking a scan and aligning it, using Zipper (Section 4.1.2), with the previously scanned and assembled geometric model.[7] By this two-link chain of transformations, from object space through scanner space to camera space, we established the positions of the light source and camera relative to the object.

To have radiometric calibration relating the digital readout of the camera to meaningful radiance measurements, we assumed uniform flat field response for the camera and an angularly uniform illumination field for the light source (shown to be valid by experiments with the same type of lens and the same light source in Appendix D) and established the single scale factor required to account for camera responsivity and light source intensity by photographing a calibrated diffuse white reference sample. This sample was included in every photograph, because the light source intensity varied considerably from one flash to the next.

To avoid having to scan the object for every photograph, we placed the object on the scanner's turntable, scanned it once, then used the turntable to precisely rotate the object to several positions while it remained fixed relative to the turntable. This led to sets of camera positions arranged on circles around the object: a series of photographs, separated

[7]In some cases we refined Zipper's estimates using a few manually specified point correspondences, gaining a mild improvement in registration.

by turntable rotations, led to one circle; the object was repositioned on the turntable to expose previously hidden surfaces, and another series of photographs was taken, forming a second circle. This process of repositioning the object, scanning it, then taking a series of photographs around the circle was repeated as often as necessary—normally resulting in three or four circles with six to eight exposures per circle (Figure 4.10).

4.6 Results

We put the algorithm described in Section 4.3.1 into practice using the setup described in Section 4.5. We scanned and built texture maps for two complex natural objects: a jagged piece of rock and a colorful squash.

The first object, a rock 23 cm in length with mixed composition, has a very complex surface texture but exhibits little noticeable specularity, so the Lambertian BRDF model is appropriate and we need not use the specular exclusion factor α_s in the algorithm. The geometric model was built using Zipper and VRIP (Section 4.1.2) from 57 individual range images, resulting in a manifold triangle mesh with 6.5×10^5 faces, which is shown rendered with homogeneous gray reflectance in Figure 4.11. The parameterization process broke the surface into 42 texture patches, which are illustrated in Figure 4.12. Some of the 16 photographs are shown in Figure 4.13. The resulting texture maps are shown in Figure 4.14; areas where the accumulated weight was zero are colored blue. A rendering of the texture mapped model and a photograph taken under similar conditions are shown in Figure 4.15.

One objectionable feature of the resulting color maps is that they are blurred and contain multiple images. This is caused by averaging misregistered samples of the surface texture; the misregistration can be caused by inaccuracies in either the geometry or the camera pose.

The second object, a multi-colored squash about 12 cm in diameter, has fairly uniform specular reflection across most of its surface. There are significant variations from this uniformity in a sunken diffuse patch at the stem end, a diffuse protrusion at the blossom end, and several small blemishes in the surface elsewhere. The model, which was built from 22 range images, contained 4.2×10^5 triangles and is shown in Figure 4.16. The surface was divided into nine patches (Figure 4.17) and photographed 24 times (Figure 4.18). The resulting textures are shown in Figure 4.19; for this simpler object there are almost no areas that were not observed sufficiently. The resulting rendering is compared to a photograph in Figure 4.20.

The color difference between the photograph and the rendering is due to a difference in light sources. The xenon strobe used to take the photographs for the texture maps (spectrum shown in Figure D.2), with its strong blue component, is well matched to the spectral response curves of the DCS 420, which has very weak blue response, but the

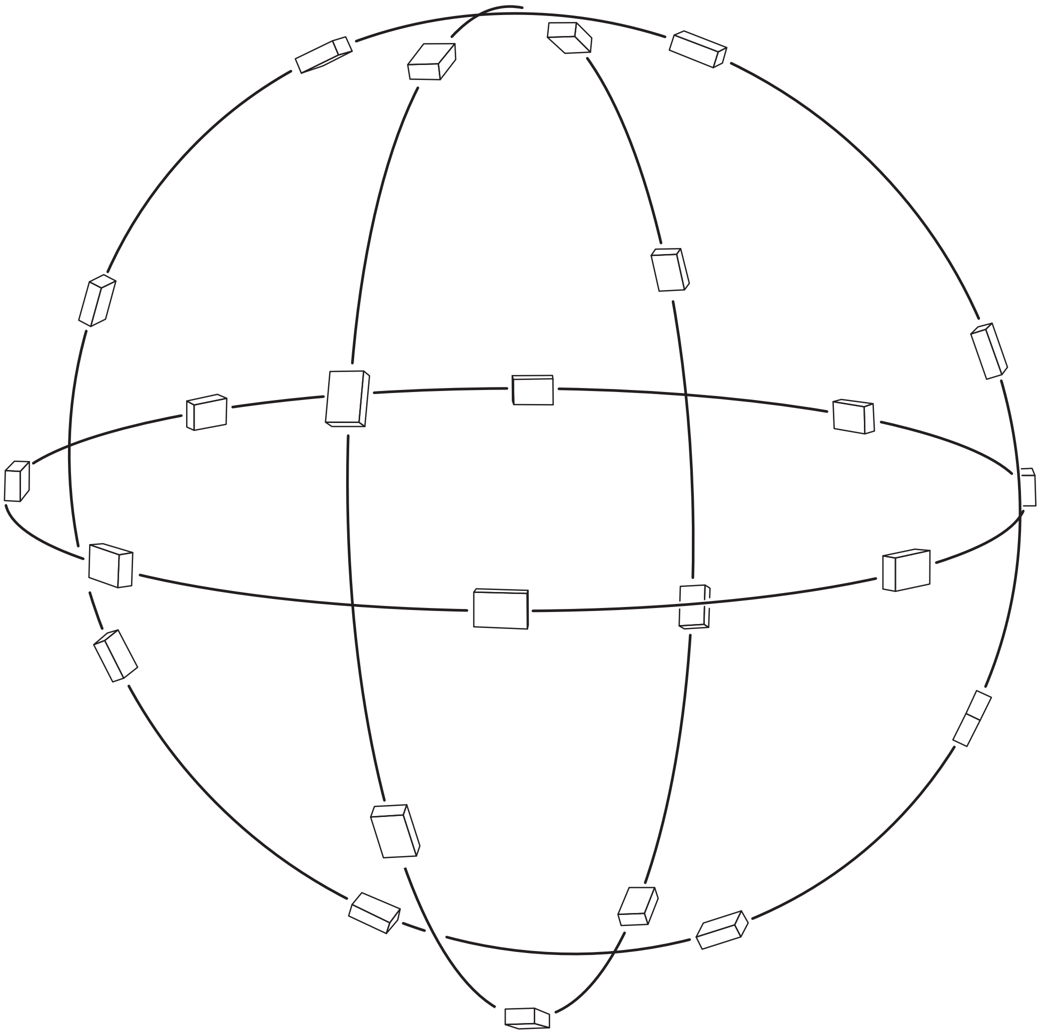


Figure 4.10: A typical set of camera positions for texture measurement. Each circle of cameras results from photographing the object at several different turntable rotations while the object remains stationary on the turntable. Picking the object up off the turntable and repositioning it leads to another circle.

tungsten light source used to take the comparison photograph has very little blue, which compounds the camera's weakness and leads to very poor color rendition.

The specular down-weighting term was used for this example, and it successfully eliminated the effect of specular highlights in the resulting texture maps. Again, however, there is some evidence of misregistration in the textures.

We applied the technique of fitting a spatially uniform BRDF model to the residuals of the diffuse estimate to account for the specular reflection, which is clearly missing from Figure 4.20. Using the isotropic version of Ward's model [64], we obtained the parameters used to render the image in Figure 4.21. Note that the specular highlights match the photograph reasonably well, though the small-scale inaccuracies in the model prevent a perfect match.

4.7 Future Work

In Section 4.3.2 we explained why not enough data are available data to measure parameters describing specular reflection independently at every surface point in the same way we were able determine the diffuse reflectance. We must somehow use data from neighboring parts of the surface to make a reasonable guess where it is impossible to make an independent estimate. We propose to do this by using a regularization term to encourage the parameters to vary smoothly across the surface when they cannot be determined completely at each point. Conceptually, we want an algorithm that will choose the smoothest function that fits the available data; this will result in a solution that is determined by the data where the right data are available and interpolates smoothly across regions where the data do not fully constrain the answer. In actual practice, there is no sharp dividing line where the parameters become underconstrained; instead, the fitting equations become increasingly ill-conditioned as the measurements become less suitable for determining the parameters. By simply adding a regularization functional to the quantity we minimize for the least squares fit, we can let smoothness take over gradually as the data cease to constrain the parameters.

We start with the previous minimization functional from Equation 4.1. Solving that equation separately for all $\mathbf{p}$ is equivalent to solving $\min_a F(a)$ where

$$F(a) = \sum_{\mathbf{p}} \sum_{k=1}^{n(\mathbf{p})} \lambda(\mathbf{p})_k^2 (f_m(a(\mathbf{p}); x(\mathbf{p})_k) - y(\mathbf{p})_k)^2. \tag{4.2}$$

The sum over $\mathbf{p}$ runs over all sample points in the texture map. Since no terms in the outer sum refer to more than one texture sample point, the sum is minimized by the parameters that minimize the inner sum at each $\mathbf{p}$. To introduce regularization, we add the following term, a scaled estimate of the L_2 norm of the first derivative of a:

$$S(a) = \sum_{\mathbf{p}} \sum_{\mathbf{q}\,\mathrm{adj}\,\mathbf{p}} \|D_\mu(a(\mathbf{q}) - a(\mathbf{p}))\|^2 . \tag{4.3}$$

Here μ is a p-vector of weights controlling the importance given to smoothness in each of the different BRDF parameters, and D_μ is a diagonal matrix with μ along the diagonal. The expression $\mathbf{q}\,\mathrm{adj}\,\mathbf{p}$ means $\mathbf{q}$ and $\mathbf{p}$ are adjacent grid points in a texture domain.

Adding this functional complicates the task of solving the system because it can no longer be treated separately at each point. It is a large, sparse, nonlinear least-squares problem with tens to hundreds of thousands of variables. It is sparse because each variable only interacts with a few other variables: F connects all the parameters at a particular data point, and S connects the same parameter at neighboring points. This lends a structure to the matrix problems that arise from these equations, which we must exploit to have a tractable algorithm. To put the reflectance estimation system into a more compact form, let $g(a) = F(a) + S(a)$.

The widely used Gauss-Newton and Levenberg-Marquardt methods [49] for nonlinear least squares problems require the solution of square linear systems with the symmetric matrix $g'(a_c)^T g'(a_c)$, where a_c is the estimate of the parameters at the current step in the iteration.[8] This matrix is non-zero at entry (i, j) only if variables i and j interact with one another. Consider a single texture map with n_y rows and n_x columns. If we order the variables first by row, then by column, then by parameter (that is, the parameters for a particular point are grouped together, and the points are listed in English reading order), then the structure of $g'(a_c)^T g'(a_c)$ is as diagrammed in Figure 4.22. Since all the parameters at a point can interact through F, there are $p \times p$ blocks of nonzero entries along the diagonal. The horizontal connections between neighbors lead to entries on the p^{th} diagonal,[9] and the connections between vertical neighbors lead to entries on the $(n_x p)^{\mathrm{th}}$ diagonal.[10] To solve this sparse linear system, we turn to the method of conjugate gradients [25], an iterative algorithm that solves linear systems with only the need to multiply by the matrix—a computation that is quite efficient for a sparse matrix like ours. The conjugate gradient algorithm can be further accelerated by using a *preconditioner*, an approximation to the system that can be solved efficiently by direct means. We use a variant of the incomplete block Cholesky preconditioner described by Golub and Van Loan [25].

[8] $g'(a)$ is the derivative of g (sometimes called the Jacobian) evaluated at a.

[9] The p^{th} diagonal of a symmetric matrix consists of all the entries (i, j) with $|i - j| = p$.

[10] This form of matrix also arises when solving partial differential equations on rectangular grids.

4.8 Conclusion

We have presented a framework for the broad problem of measuring spatially varying reflectance characteristics of complex surfaces. The process begins with a geometric model of the surface, along with measurements of reflected radiance under known lighting conditions. So that the results can be represented using raster texture maps, we define a number of overlapping, parameterized texture patches that cover the surface. We then consider all the pixels in all the photographs as measurements to which spatially varying BRDF parameters must be fit.

If the BRDF model has no rapid directional variations, each pixel of each texture map can be processed independently, leading to a simple, efficient solution. We have demonstrated this technique, using a Lambertian model, on complex objects with detailed textures. The results capture the textures well, allowing photorealistic rendering.

If the BRDF model includes a specular lobe, it no longer suffices to consider each point separately. This is because information about the specular lobe can only be observed from near-specular configurations, and most surface points will not happen to fall within specular highlights in enough images to provide the data needed to solve for the specular parameters of the BRDF model. To solve this problem, we propose the introduction of a regularization term that encourages smoothness across areas where parameters are underdetermined.

An extreme case of this idea is to assume that specular parameters are constant across the surface while diffuse parameters are variable. This leads to another quite practical solution, which we have demonstrated on real data.

In the more general case, in which all p parameters vary across the surface, we have explained why it is not practical (or even possible for most objects) to solve for the complete BRDF model across the entire surface. We have proposed a way of solving a regularized system to fit the available data while interpolating smoothly where the solution is underconstrained. Compared to previous techniques for handling spatially varying specular reflection, our proposed approach would provide a better-founded and more general way to take advantage of the incomplete information that is available.

Figure 4.11: The geometry of the rock scan alone, with no texture map.

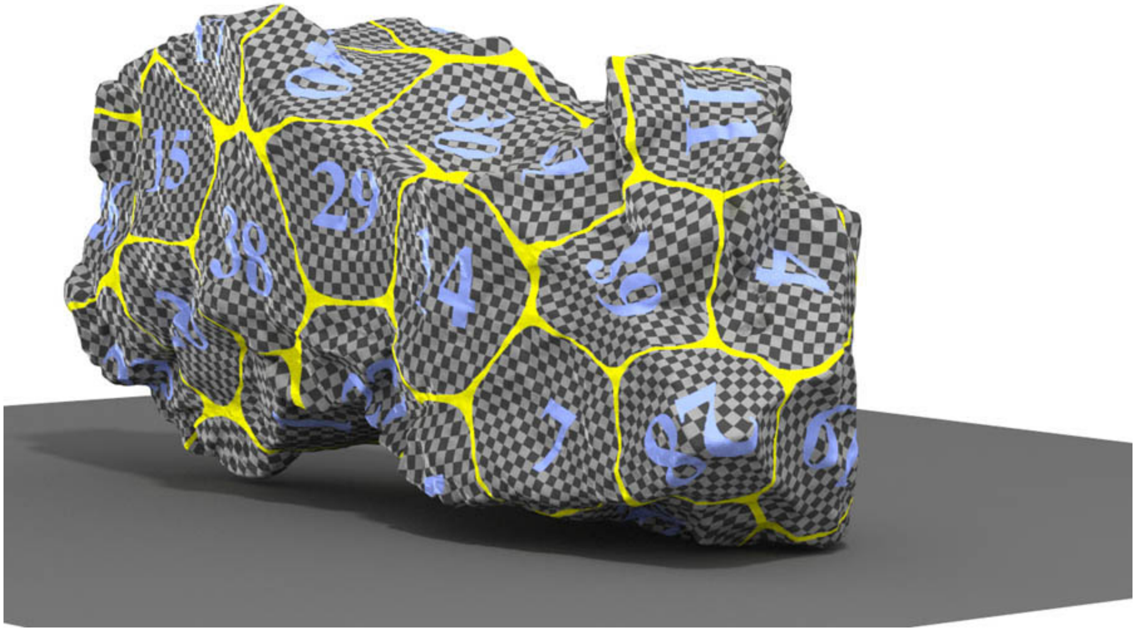


Figure 4.12: The texture patches on the rock model. The model has been rendered using a checkerboard with a blue number and a yellow border for each of the texture maps to illustrate the locations of the patches. Note that the checkerboards do not appear very distorted, which demonstrates the low distortion of the texture embeddings.

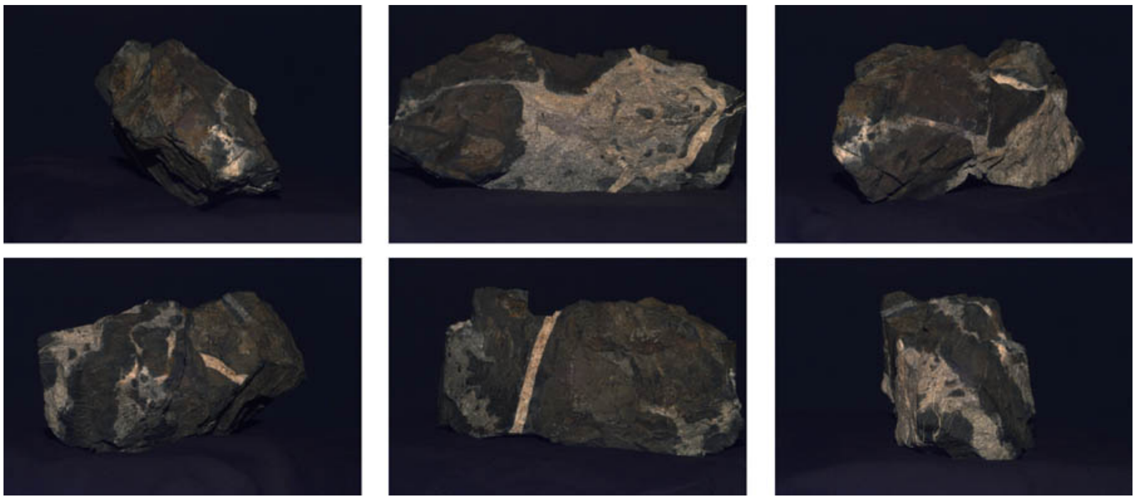


Figure 4.13: Some representative photographs from the set of 16 used to compute texture maps for the rock.

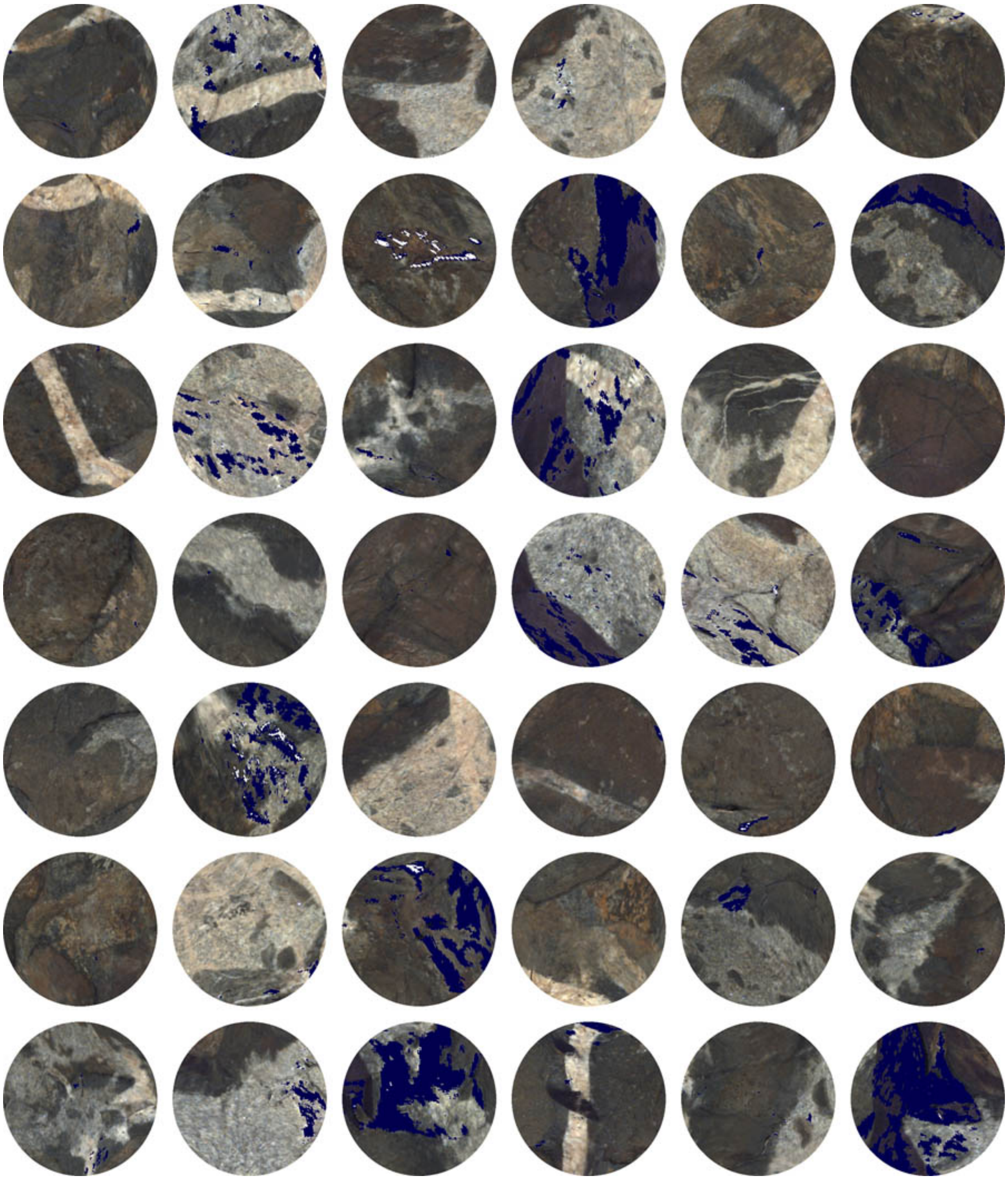


Figure 4.14: The 42 texture maps computed for the rock model.

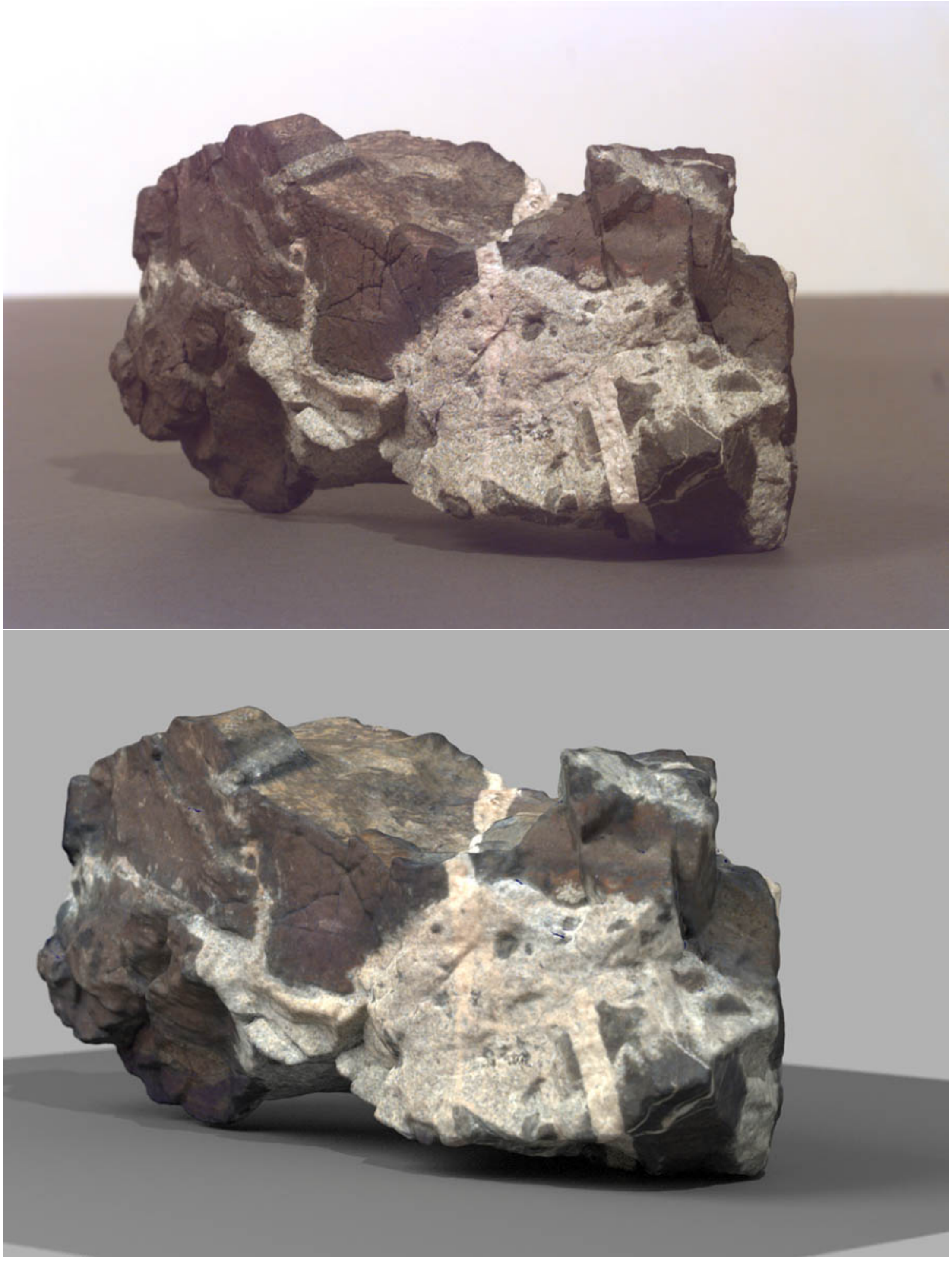


Figure 4.15: A comparison between the textured rock model and the actual rock. Above is a photograph of the rock, and below is a rendering of the model under similar lighting conditions. The difference in color is due to the tungsten light source used to take the photograph.

Figure 4.16: The geometry of the squash scan alone, with no texture map.

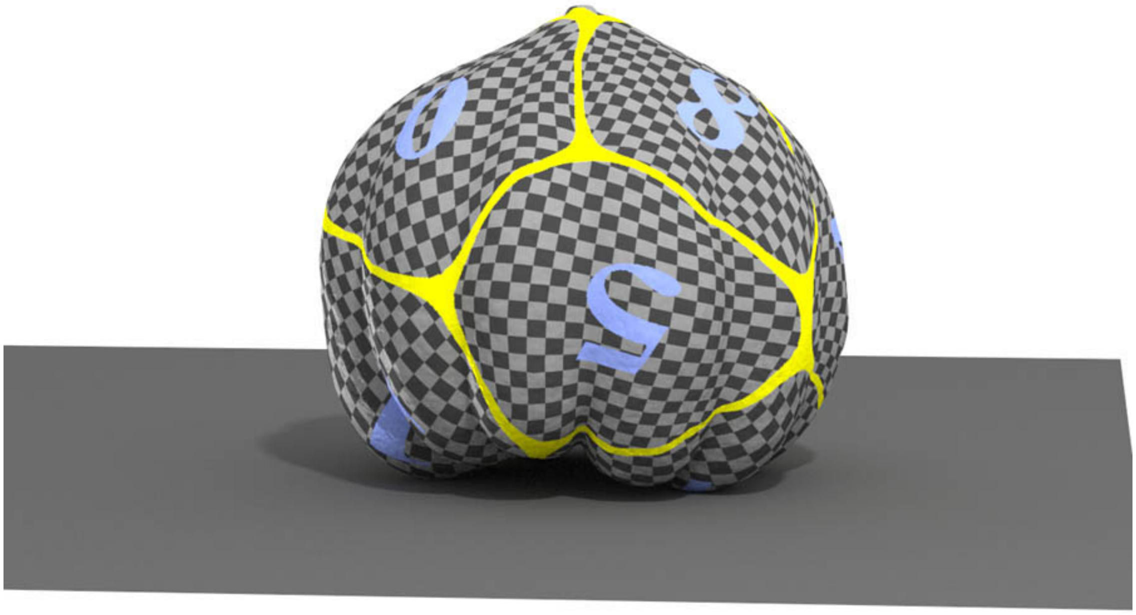


Figure 4.17: The texture patches on the squash model. The model has been rendered using a checkerboard with a blue number and a yellow border for each of the texture maps to illustrate the locations of the patches. Note that the checkerboards do not appear very distorted, which demonstrates the low distortion of the texture embeddings.

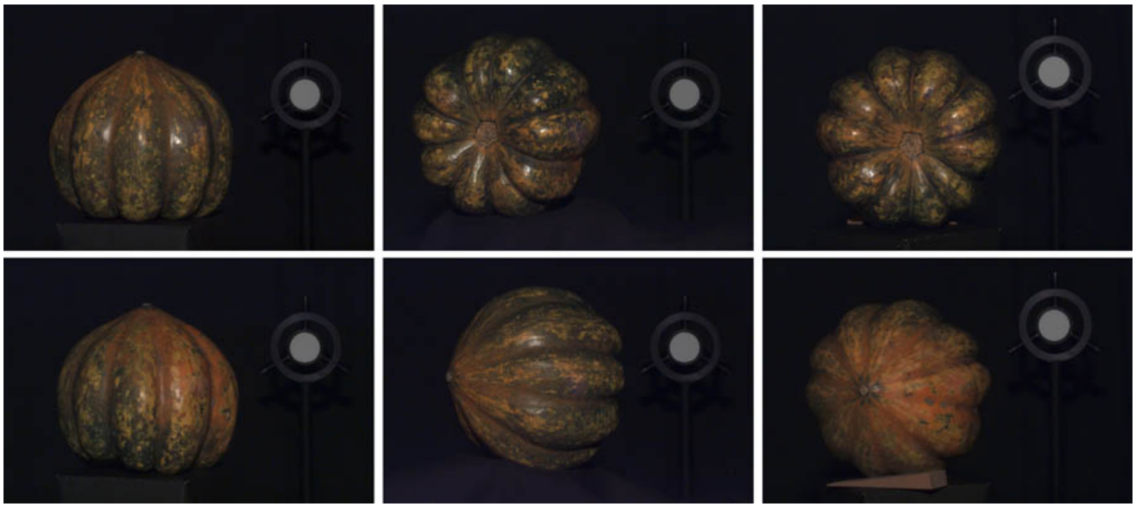


Figure 4.18: Some representative photographs from the set of 24 used to compute texture maps for the squash.

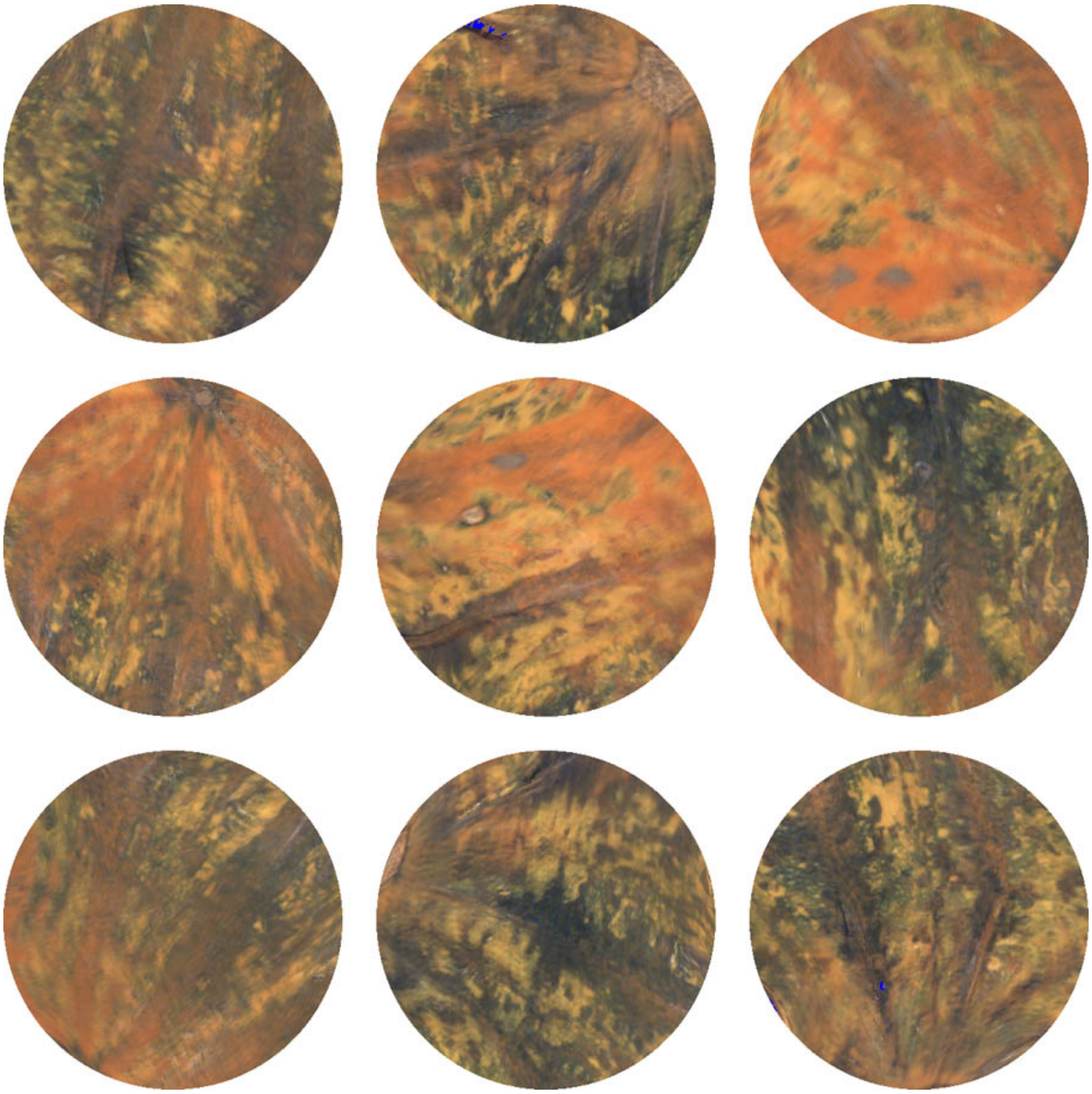


Figure 4.19: The nine texture maps computed for the squash model.

Figure 4.20: A comparison between the textured diffuse squash model and the actual object. Above is a photograph of the squash, and below is a rendering of the model under similar lighting conditions. The difference in color is again due to the tungsten light source used to take the photograph. Because we have only measured the diffuse component, the highlights are missing from the rendered image.

Figure 4.21: The squash model with the fitted spatially uniform specular component.

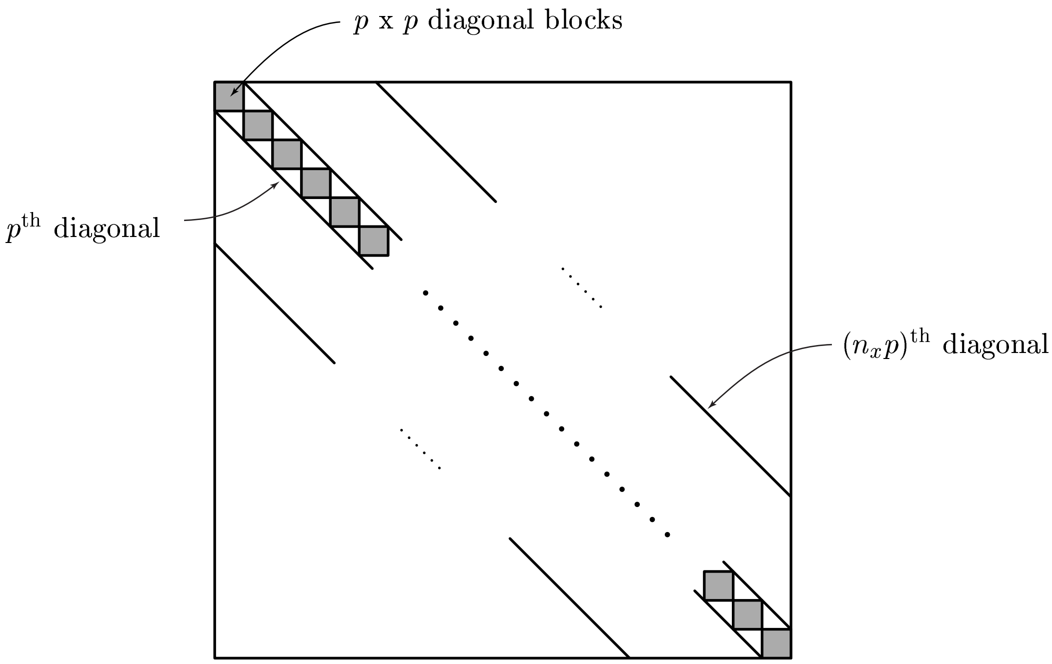


Figure 4.22: The matrix structure of the linear subproblem at the core of the proposed regularized reflectance estimation system.

Chapter 5

Image-based BRDF Measurement

In Chapter 4 we considered the problem of measuring an object's surface texture using photographs. In order to allow for inhomogeneous surface reflectance, we had to make quite stringent assumptions about the form of the BRDF. In this chapter we will work under the opposite circumstances: we will assume that the surface is homogeneous, but we will make no assumptions at all about the form of the BRDF, other than that it is isotropic. Our goal will be to measure the full BRDF in as much detail as possible.

5.1 Overview of Method

A straightforward device for measuring isotropic BRDFs is illustrated in Figure 5.1. A flat sample of the material to be measured is illuminated by a light source, and a detector measures the complete distribution of reflected light by moving around the entire hemisphere. In this way the complete scattering behavior for a particular angle of incidence is measured. This process is repeated many times, moving the light source each time to measure a different incidence angle. In this way the whole BRDF is measured. We will refer to this device as Device A. The light source only needs to change its elevation, from normal to grazing; its azimuth can remain fixed because the BRDF is assumed to be isotropic. Because there are three dimensions to the domain of an isotropic BRDF, there are three mechanical degrees of freedom in Device A: two for the detector and one for the source.

Because the positions of the light source and detector are only relevant relative to the plane of the surface, exactly the same results could be achieved using a rotating sample and a fixed detector, as shown in Device B (Figure 5.2). There are still three degrees of freedom: two for the sample and one for the source. The ability to change the sample's orientation substitutes for the ability to change the absolute direction to the detector.

If the sample is curved, instead of flat, every part of the sample's surface has a different orientation. If the sample curves sufficiently to include all the necessary orientations, we

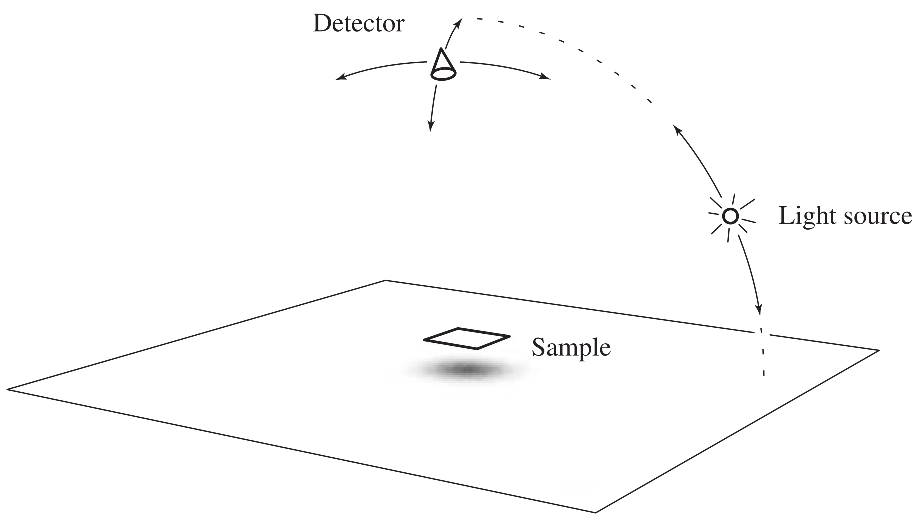


Figure 5.1: Device A for measuring BRDFs. The source moves to change the incident angle, and the detector moves to change the exitant direction.

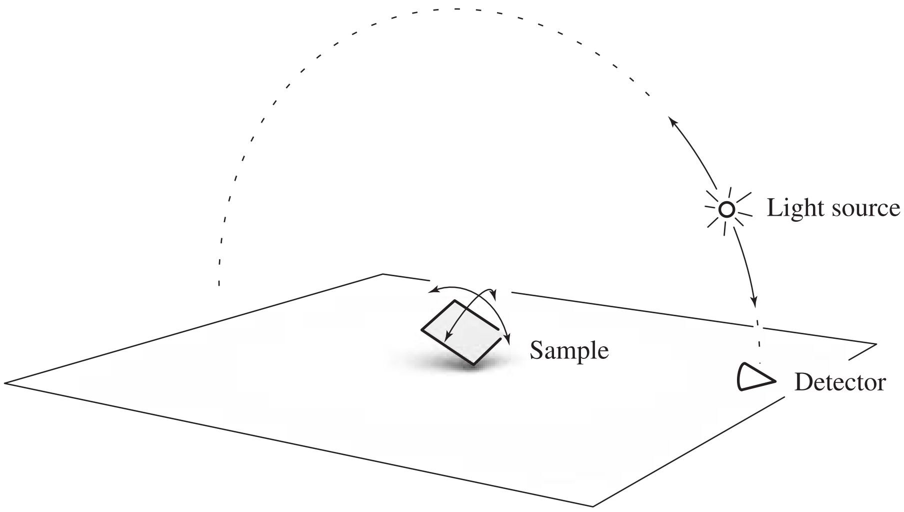


Figure 5.2: Device B for measuring BRDFs. The source moves to change the incident angle, and the sample tilts to change the exitant direction.

can obtain the same measurements as Device B by measuring different parts of the surface instead of rotating the sample. This is the key observation for image-based BRDF measurement. We can measure all parts of the surface at once by using a camera for the detector (Figure 5.3). There are still three degrees of freedom: one for the light source and one for each dimension of the camera's image.

The major advantages of this approach are speed and sampling density. Rather than having to move the detector to each of hundreds of positions, pausing each time to make a measurement, we can capture hundreds of thousands of samples in a single exposure using a high-resolution image sensor. Only a single exposure is required for each light source position. An attendant disadvantage is that it is no longer easy to capture the full wavelength spectrum of the reflected light. For computer graphics great detail in the spectrum is normally not required, and we can gather enough information by measuring sequentially through a small number of color separation filters.

5.1.1 Sampling patterns

Collecting data in this way leads to very different sampling patterns than we normally expect from a BRDF measurement device. If we consider the set of BRDF configurations that is measured by an image, we find that each image creates a curved *sheet* of measurements in the BRDF's domain. In the absence of occlusion, the curvature of a smooth surface creates a continuous function from image position to the domain of the BRDF, embedding the image as a two-dimensional surface in that domain. The geometry of the surface, light source, and camera dictates where that sheet falls and how the samples will be arranged on it. This leads to challenges in understanding the form and arrangement of the sheets, and in reconstructing BRDF values or regularly spaced samples from this irregularly structured data.

A two-dimensional example is illustrative, both in understanding the technique and in testing the implementation. If the object to be photographed (the *test sample*) is a cylinder, and we restrict our attention to a plane perpendicular to its axis and containing the camera and light source (Figure 5.4), we can measure incidence-plane BRDFs, using one-dimensional sheets to cover the two-dimensional domain. As can be seen in Figure 5.5, the angle between the viewing and illumination directions remains approximately constant within each image, to the extent that the camera and light source are far away compared to the size of the cylinder. This leads to an approximately constant difference between θ_i and θ_e, which means that the measurement sheets, plotted against θ_i and θ_e, will approximate straight lines at $45°$ to the axes. Each measurement image leads to one sheet of measurements, with its position determined by the relative positions of the camera and light source. By moving the light source, we can run $\theta_i - \theta_e$ through the full range from near zero, with

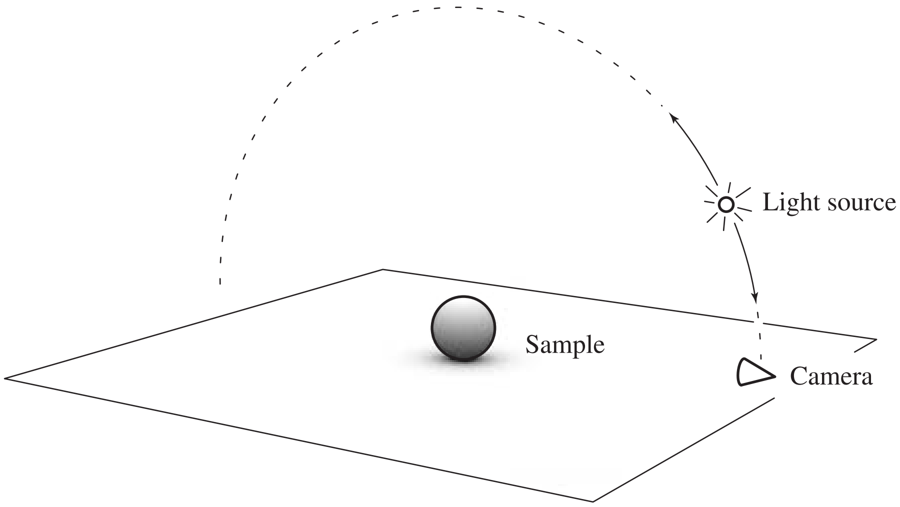


Figure 5.3: An image-based BRDF measurement device. The source moves to change the incident angle, and the sample's curvature allows all exitant directions to be measured simultaneously.

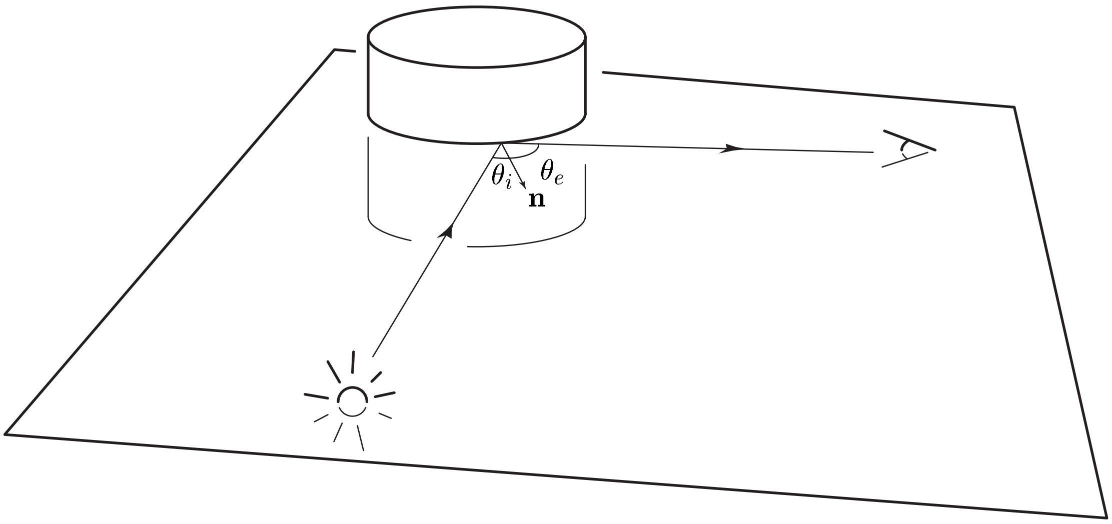


Figure 5.4: Measuring incidence-plane reflection from a cylindrical sample. With the light source, surface normal, and camera all in a plane, each image measures the BRDF for a range of values of θ_i and θ_e.

the light source next to the camera, to 180°, with the light source behind the sample. The physical shape of the camera and light source limits the approach to retroreflection,[1] and the size of the light source and camera aperture limits the approach to grazing reflection in the incidence plane. Figure 5.6 shows the sample sheets for an actual experiment using a cylindrical sample; note that the sheets are very nearly 45° lines.

In the case of a full isotropic BRDF measurement, we substitute a sphere for the cylinder, and we use the full two-dimensional image. With exactly the same sequence of light source positions, we then obtain a series of two-dimensional sheets, which together fill the three-dimensional domain of an isotropic BRDF. The angle between the incident and exitant directions is again nearly constant, but the geometric shape of the sheets depends on how the BRDF's domain is parameterized. Normally it will not be as simple as the approximate lines of Figure 5.6. We return to this issue in Section 5.6.

The following sections will describe the apparatus used to make our image-based BRDF measurements, its use and calibration, and the issues involved in understanding and processing the data that it produces. A detailed description of the procedure used to make these measurements is given in Appendix F. We concentrate entirely on isotropic materials, although the system can be extended to measure anisotropic BRDFs as well.

5.2 Prior Work

The BRDF is a function of five variables, if wavelength is included, although for isotropic materials there are just four degrees of freedom. Sampling this high-dimensional space sequentially is impractical, but measuring multiple points simultaneously can speed data collection.

In a classical setup [45, 59, 65], the three or four angular dimensions are handled by specialized mechanisms that position a light source and a detector at various directions from a flat sample of the material to be measured. The final dimension, that of wavelength, is handled either with a broadband spectroradiometer that measures the entire spectrum at once, or by multiple measurements varying the wavelength of a narrow-band source or detector. Because three, four, or five dimensions must be sampled sequentially, measuring reflectance functions can be time-consuming, even with modern computer controls. Moving the motor stages and measuring the reflected light can take several seconds, and since measurements are taken point by point, even a sparse sampling of the incident and exitant hemispheres can take several hours.

[1] As is often practiced in reflectance measurement, we could measure angles up to exact retroreflection by using an angled beam splitter to allow the source and camera to share the same beam without physically colliding.

More recently, techniques have been reported in the computer graphics literature to reduce dimensionality in angle rather than in wavelength. These methods, like the method presented in this chapter, use a two-dimensional detector—the image sensor of a digital camera—to measure a two-dimensional range of angles simultaneously, leaving one or two dimensions of angle and one dimension of wavelength to be sampled by sequential measurements.

Ward [64] presents a device to measure the BRDF of anisotropic materials by using a hemispherical half-silvered mirror to gather light scattered from a flat sample into a CCD camera with a fish-eye lens. The camera thus captures the entire exitant hemisphere at once for each illumination direction, leaving two degrees of freedom to handle mechanically. This provides significant time savings over the four degrees of freedom required by the conventional approach. Ward's instrument is limited by its optics; the hemispherical mirror only approximates the ideal ellipsoid, and vignetting limits the quality of measurements near grazing exitance. The device integrates energy over the entire visible spectrum; it does not measure variation in the BRDF with wavelength, although sequential measurements through color filters could be added.

Karner et al. [36] describe a system using an inexpensive CCD camera and a simple incandescent lamp. In this case, the camera captures an image of a large flat sample and a flat reference surface, which are illuminated symmetrically by the small light source. The different points on the samples have different illumination and reflection directions; because of the symmetry of illumination, the BRDF values can be computed from the ratios between corresponding pixels on the two samples. This method, like Ward's, handles two dimensions of angle by simultaneous measurement, but the authors do not try to sample the entire BRDF, because their goal is to fit a simple reflectance model rather than to measure the full BRDF. They measure wavelength dependence using the built-in color filters of the CCD camera.

Ikeuchi and Sato [34] present a system for estimating reflectance model parameters using a surface model from a range scanner and a single image from a video camera. In contrast to the methods of Ward and Karner et al., they use a curved sample to capture a set of directions spanning a large range of both incidence and exitance angles. Because their goal, like that of Karner et al., is to fit a reflectance model, they use a single image and make no attempt to sample the BRDF exhaustively.

Sato et al. [54] describe a method to fit BRDF parameters from a sequence of images of an arbitrarily shaped object under controlled illumination. They use the two dimensions of the captured images to capture the spatial variation of BRDF across the surface, rather than to sample angular parameters of a spatially uniform BRDF. The image sequence provides samples along a one-dimensional path for each surface point; a simple reflectance model is fit to these data. Again the goal is model fitting, so only a small portion of the BRDF parameter space is measured.

Like these other image-based systems, the system presented in this chapter uses a camera to sample a two-dimensional set of angles in a single measurement, so it shares their advantages in speed and sampling density over traditional approaches. Our method, however, measures isotropic BRDFs very completely, so the data are useful for more than fitting low-dimensional models. Our results can be used not only to render images, but also to validate reflectance models for particular materials, or to investigate BRDFs that do not conform to existing models. We have also verified the accuracy of our system more thoroughly than previous reports of image-based methods.

5.3 Apparatus

Our photographic BRDF measurement technique requires a well-characterized camera, a stable and uniform light source, and a means for measuring their positions. Also required are curved samples of accurately known shape.

The system we used is shown in Figure 5.7. The main parts are the *primary camera*, which takes the photographs from which the measurements are made, a light source, the test sample, and the *secondary camera*, which is used to measure the position of the light source.

The measurement of the light source position is a novel aspect of our technique. The source was mounted rigidly to the secondary camera, and its position was found by first determining the position and orientation (the *pose*) of that camera. Each measurement exposure was made by opening the shutter of the primary camera, then triggering the secondary camera during the exposure. The secondary camera triggered the flash, so we obtained a calibration image directly correlated with the source position, acquired at exactly the same time as the measurement image. A number of machine-readable targets with known 3D positions were placed near the sample, so that each calibration image included images of several targets. By analyzing these images, the poses of the secondary camera were determined. With the light source rigidly attached to the camera, its position was easily found for each exposure. The algorithms used to generate and recognize the targets, establish their 3D positions, and determine the poses of the secondary camera are described in Appendices B and C. The light source (a xenon flash) and the secondary camera, a Kodak DCS420, are described in Appendix D.

Because this technique can locate the light source equally well at any location where the secondary camera can see the targets, it gives us great freedom in placing the light source. We chose to move the source manually from one position to the next, using a path marked on the floor as a guide to approximate placement. We used a path shaped like a half-turn of a spiral (Figure 5.8 shows the actual path from a typical measurement) so that the light source distance increased as the configurations approached grazing. This automatically reduced the signal level for these very bright reflections while increasing the density of sheets in this fast-changing region of the BRDF.

5.3.1 The primary camera

The primary camera was a Photometrics PXL 1300L; it is described in Appendix D. It remained fixed throughout the measurement process and made the actual measurements of radiance reflected from the test sample. Its overall sensitivity was adjusted, both by adding and removing neutral density filters from its optical path and by adjusting the lens aperture, as appropriate to allow measurement of bright reflections without saturation. If a wavelength-dependent measurement was required, we made sequential exposures using color separation filters. We also used an infrared blocking filter to eliminate unwanted invisible light.

The physical setup of the primary camera and the related optics is diagrammed in Figure 5.9 and shown photographed in Figure 5.10. The filters themselves are described in Appendix D. To prevent stray light from contaminating the signal, we enclosed the lens and filters in a black box with an opening in the front just large enough for the required field of view.

5.3.2 The test samples

Our method requires accurate knowledge of the geometry of the sample's surface. In addition, to avoid shadowing, occlusion, and interreflection, the sample should be convex; and to provide a full range of BRDF samples, the surface should be smoothly curved in both directions. Two approaches to obtaining the required geometry are possible: one could begin with an arbitrary object and measure its shape using a range scanner, or one could use an object manufactured to a specific shape. We adopted the latter approach, but the first is equally viable and would require no modifications to the technique.

Our test samples consisted of cylinders and spheres; the cylinders were sections of aluminum tubing, with a nominal outside diameter of six inches. The spherical samples were 200 mm copper spheres.[2] We placed them on a Cyberware MS motion platform (Appendix E) to provide accurate computer-controlled translation and rotation. We used this

[2]We estimated the cylinders to be within 0.5 mm of round and the spheres to be within 1 mm of spherical. The spheres had small-scale flaws near the pole that introduced some minor artifacts in the data near grazing incident angles.

equipment because it was already available in our laboratory and it made some calibration simpler, but such a device is by no means required for making image-based BRDF measurements.

5.3.3 Calibration

In order to interpret a pixel value as a measurement of BRDF, the following information must be known:

- The responsivity of the pixel sensors to scene radiance;
- The irradiance due to the light source at the relevant surface point;
- The geometric arrangement of the surface normal, the viewing direction, and the illumination direction.

We undertook several steps to ensure that each of these items was well controlled. The specific calibration processes for each camera and for the light source are described in Appendix A.

As mentioned earlier, we calibrated the poses of the secondary camera using observed positions of the calibration targets, in order to locate the light source. The focal length and pose of the primary camera could have been found in the same way, but the arrangement of the apparatus was such that very few calibration targets appeared in the primary camera's field of view. To rectify this, we placed a temporary set of targets in the scene, measured their 3D positions by photographing them with the secondary camera, then used those as the known targets to calibrate the primary camera's pose and focal length.

We established the sample's horizontal position by centering it on the turntable (Appendix F). Because we did not measure the height of the sample above the table, we computed it from the measured center of the sample's image in the primary camera. The measured sample radius was less precise than the other measurements, so we made an adjustment to bring the target into exact agreement with its image in the primary camera. The most important consideration for accuracy is that the silhouette edge of the sphere's image agree with the model; whether it is brought into agreement by adjusting the focal length or the sphere's radius makes an insignificant difference. Because it was easier, we adjusted the focal length of the primary camera.

For the actual measurements, and after verifying that it would not introduce any significant errors, we further altered the primary camera parameters so that the camera's optical axis passed through the center of the test sample, in order to simplify subsequent computations.

5.4 Data Processing

The most important data that result from the measurement of a single BRDF are:

- A set of measurement images from the primary camera, one for each light source position (three for each position if the color separation filters were used), with a record of the lens aperture and neutral density filter used for each exposure.
- A set of light source calibration images from the secondary camera, one for each light source position.

The basic steps to process these data are as follows.

1. *Find the poses of the secondary camera.* First, the calibration images are analyzed to locate and identify all the visible and legible targets in each image. Then, the pose estimation algorithm is used to compute the pose of the secondary camera at each light source position from the observed target locations and the targets' previously measured 3D positions. The only user intervention required in this step is to remove any mistakenly recognized targets from the output of the target finder; images with such errors are automatically brought to the user's attention using a threshold on the residual error in the pose estimation equations.
2. *Locate the image of the test sample in the primary camera.* The user identifies several points along the silhouette of the sample in one of the photographs, and these points are used to fit either a circle, in the case of a spherical sample, or two parallel lines, in the case of a cylindrical sample, which describe the exact position of the sample in image coordinates.
3. *Compute the BRDF samples.* For many points in each image, trace the corresponding ray from the camera, find its intersection with the test sample, and compute the directions ω_i and ω_e and the surface normal $\mathbf{n}$. Compute the relative radiance using the calibration information about the primary camera, and compute the relative irradiance from $\langle \omega_i, \mathbf{n} \rangle$ and the distance to the light source. The ratio of those numbers is a measurement of $f_r(\omega_i, \omega_e)$.

The set of image points where the BRDF samples are computed in the last step can be different, depending on the pattern of samples desired for the output. The simplest approach is to generate one BRDF sample for every pixel that falls within the image of the test sample. In this way, the output samples correspond one for one with the physical measurements made by the individual CCD elements. However, in some circumstances it is helpful to have sets of samples with their locations constrained in some way; for instance, we may want to generate a set of samples for a fixed incidence or exitance direction in order to make a traditional scattering diagram.

Let us take the example of generating a full set of samples for a spherical test sample and a fixed incidence direction θ_i^0. In general, none of the pixels will have $\theta_i = \theta_i^0$ exactly. However, if we consider each image as a representation of a continuous 2D function and allow ourselves to reconstruct values at arbitrary positions within the image, we can use sample points that do have $\theta_i = \theta_i^0$. Such image points can be found by projecting points on the sample surface that have the desired incidence angle into the camera image. These points are all on a circle with its center facing the light source; this circle can easily be computed, and projecting a sequence of regularly spaced points along that circle (only the ones visible to the primary camera, of course) into the image gives us the set of sample points we seek. Repeating this process for every image gives a series of rings or partial rings of samples, each corresponding to one image, that together cover the whole exitant hemisphere for $\theta_i = \theta_i^0$. These rings are, in fact, the intersections of the sample sheets with the set $\theta_i = \theta_i^0$. An example for an actual data set can be seen in Figure 5.11; Figure 5.12 shows the rings more clearly, on the projected exitant hemisphere.

5.5 Results

We have used the system described above to measure the full isotropic BRDF of three materials, shown photographed in Figure 5.13. In addition, two materials were measured in the incidence plane only. Flat samples of some of the materials were also measured independently using a gonioreflectometer that was designed and verified for accuracy within 5%. We validated the image-based measurements both by verifying reciprocity and by comparing the data from the two independent measurement systems. The new method gives results comparable in accuracy to the gonioreflectometer: consistency is excellent out to 75° incidence (or exitance) and reasonable out to about 85°.

5.5.1 Incidence plane measurement

We measured two materials in the incidence plane only, using a cylindrical sample. One was a gray primer (Krylon sandable primer #1318 "all-purpose gray"), which was sprayed directly onto the aluminum substrate. The other was an ordinary office paper (Xerox 2400DP), which was wrapped tightly around the same cylinder, using several layers to avoid any possible substrate effects.

The resulting incidence-plane measurements are samples of a function of two variables: $f_r((\theta_i, 0), (\theta_e, 0))$. The whole dataset can be presented as a unit by plotting the measured BRDF as a height field over the (θ_i, θ_e) domain, as shown for the office paper in Figures 5.14

and 5.15. Each curve in these plots represents the sheet of measurements from a single image; in total there are approximately 4000 points. Even though the paper appears fairly diffuse under casual inspection, noticeable variation can be seen. The constant BRDF of a purely Lambertian material would lead to an entirely flat graph. This surface exhibits directional diffuse reflection, with a very broad peak along the specular locus (points for which $\theta_i = -\theta_e$); for this material the specularity takes the form more of a fold or ridge than a distinct lobe (compare Figure 5.14 and Figure 5.18).

The principle of reciprocity requires this graph to be symmetric across the specular locus; that it appears to have this symmetry is a first indication that the data are reasonable. To test this symmetry more strenuously, we plotted perpendicular slices of this graph: the data for fixed θ_i against the data for the same fixed value of θ_e.[3] The resulting plots for three angles are shown in Figure 5.16. These curves agree closely out to about 85°. For an objective error measure, we computed the RMS relative error[4] between the curves, which was 1.5% over all data out to 75° and 6.0% when all points to 85° were included.

The second test on the accuracy of our data was a comparison against independent measurements of the same material. We made these measurements using a traditional gonioreflectometer, removing the stack of paper from the cylinder and clamping it against a flat plate in the gonioreflectometer's sample holder. We made measurements for fixed incidence and exitance angles in 15 degree steps, with approximately 40 samples along the variable axis for each fixed angle. The image-based data for three fixed angles are plotted against the corresponding measurements in Figure 5.17. The RMS discrepancies between these curves are 2.8% to 75° and 3.7% to 85°.

The same data was also measured for the gray primer; the corresponding plots are shown in Figures 5.18–5.20. Note that the primer exhibits a stronger specular reflection, with a more distinct, rounded specular lobe. The error measures for this material are summarized in Table 5.1.

5.5.2 Full isotropic measurements

We measured three paints for full isotropic BRDFs, using spherical samples. The three paints were the same gray primer described in the preceding section, a blue enamel (Krylon latex enamel #7205 "true blue"), and a red metallic automotive lacquer (Dupli-color #T-345 "garnet red"). To avoid problems with extremely high dynamic range, we coated the

[3]Because the primary camera is stationary, the data for fixed θ_e come from the same point in each measurement image. However, the light source moves to arbitrary angles, so the values for fixed θ_i are to be found at a different point in each image. In both cases, sampling at predefined angles requires reconstruction of a continuous image. For these plots we used linear interpolation.

[4]We computed the RMS average of the relative error between the original data points and the interpolated reciprocal data points, where the relative error between x_i and y_i is $2(x_i - y_i)/(x_i + y_i)$.

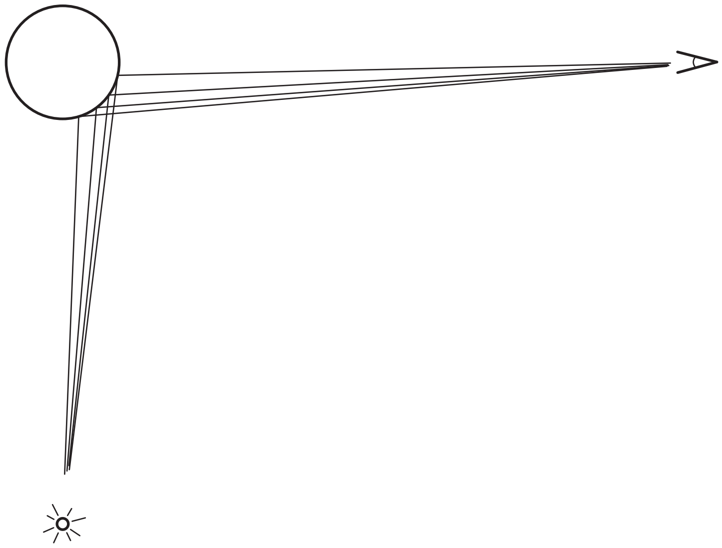


Figure 5.5: When the sample size is small relative to the distances to the camera and light source, the angle between the source and camera directions remains nearly constant for each image.

Table 5.1: Summary of error measures for several accuracy tests.

Test	RMS error to 75°	RMS error to 85°
Paper reciprocity	1.5%	6.0%
Paper against gonioreflectometer	2.8%	3.7%
Primer reciprocity	2.5%	7.8%

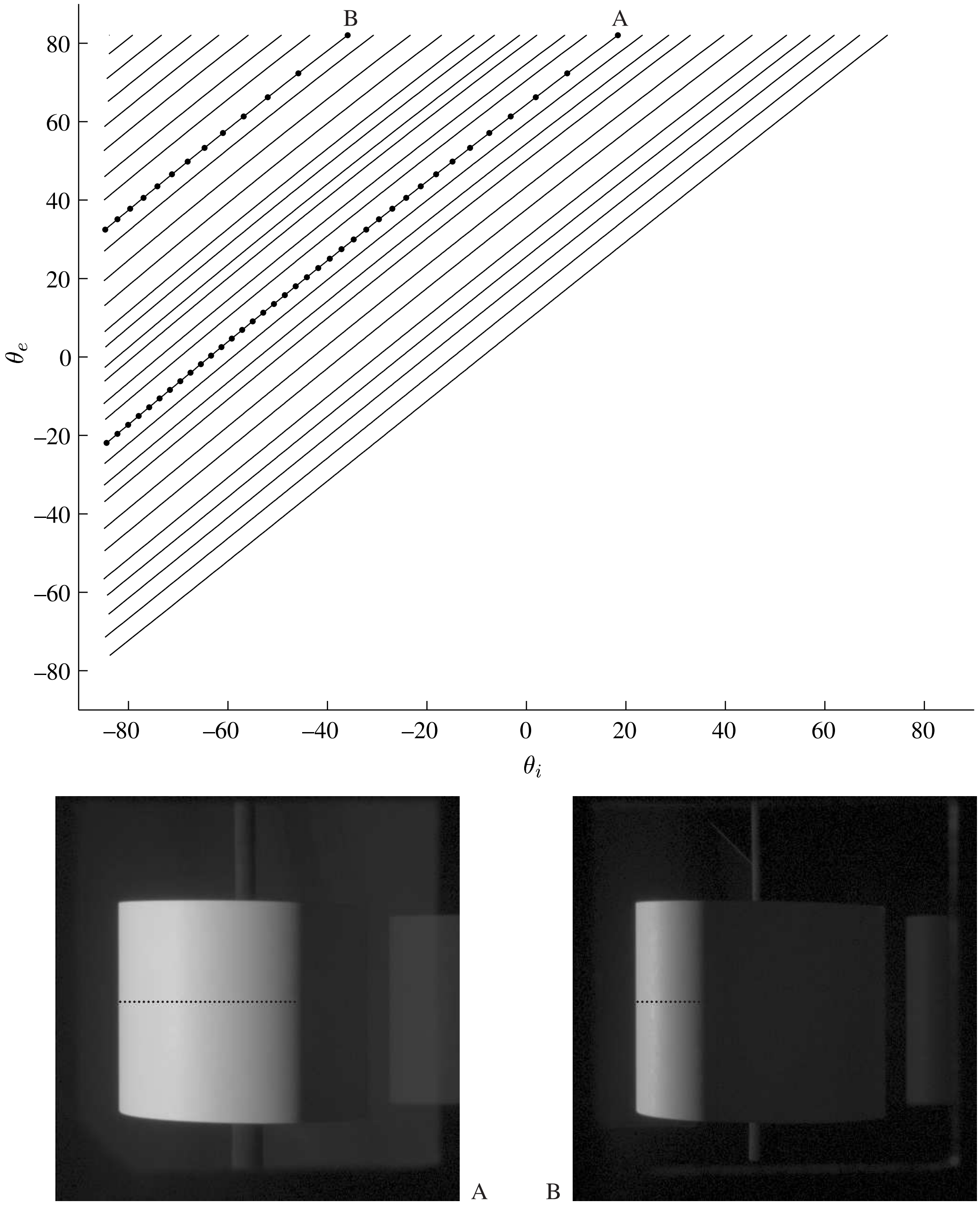


Figure 5.6: The (θ_i, θ_e) locations of the sheets from the paper measurements. Two of the actual measurement images are shown below, with the locations of several samples marked in each. The locations of the corresponding BRDF measurements are indicated by round dots in the graph above. The camera positions for these images are labeled in Figure 5.8.

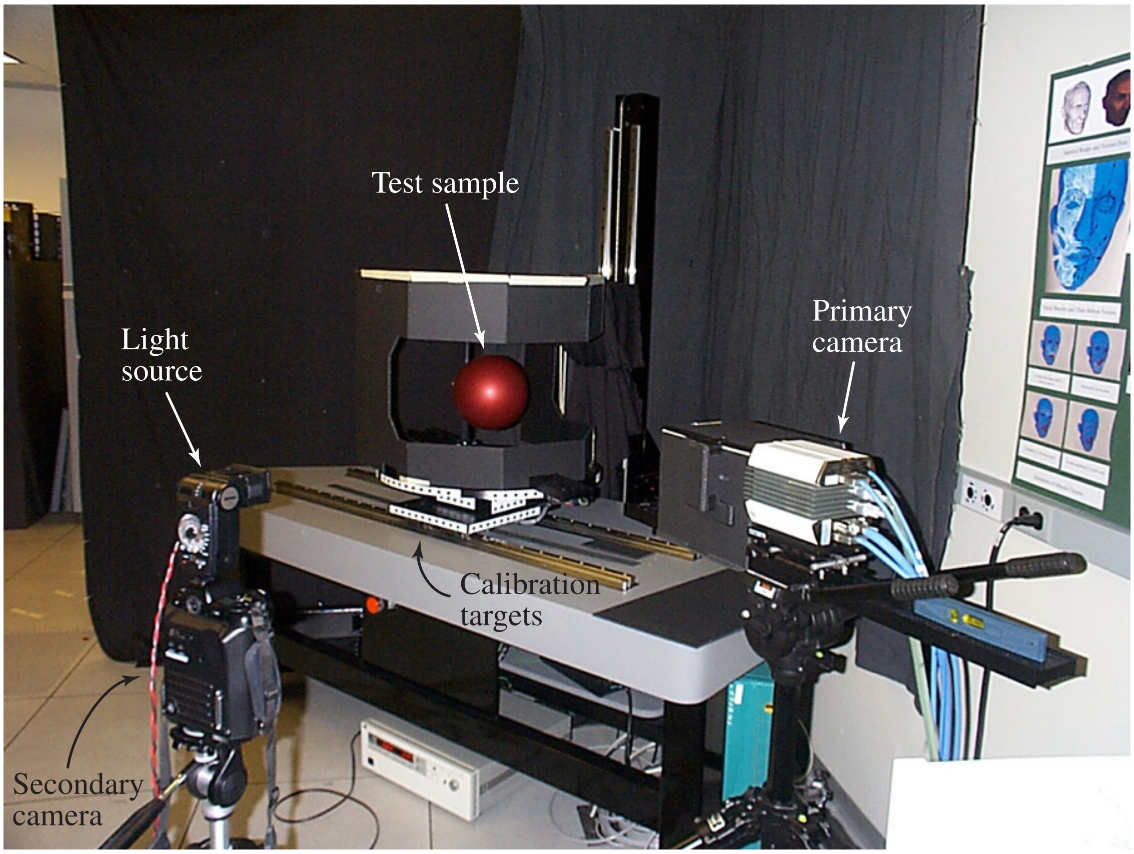


Figure 5.7: The experimental setup for image-based BRDF measurement.

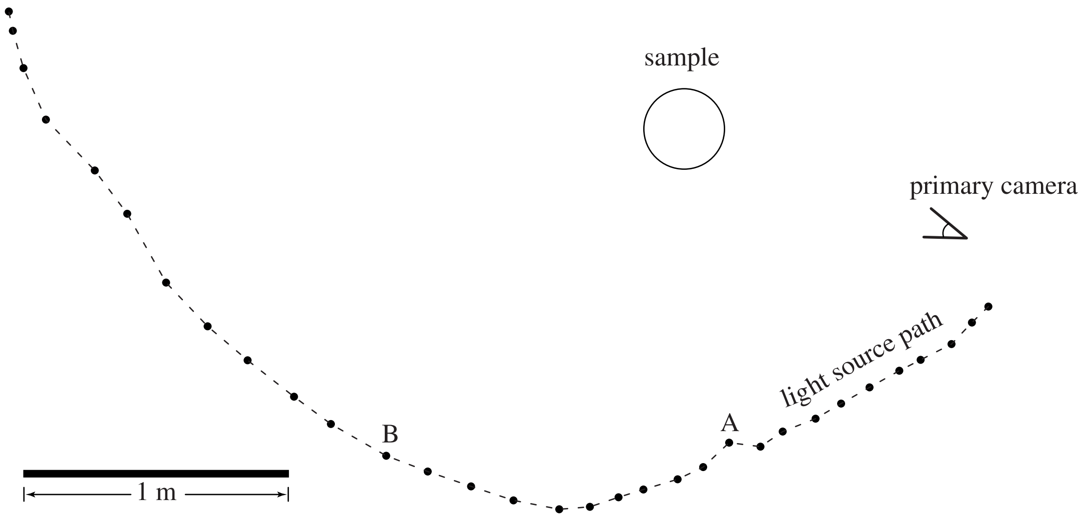


Figure 5.8: The camera path for an actual measurement.

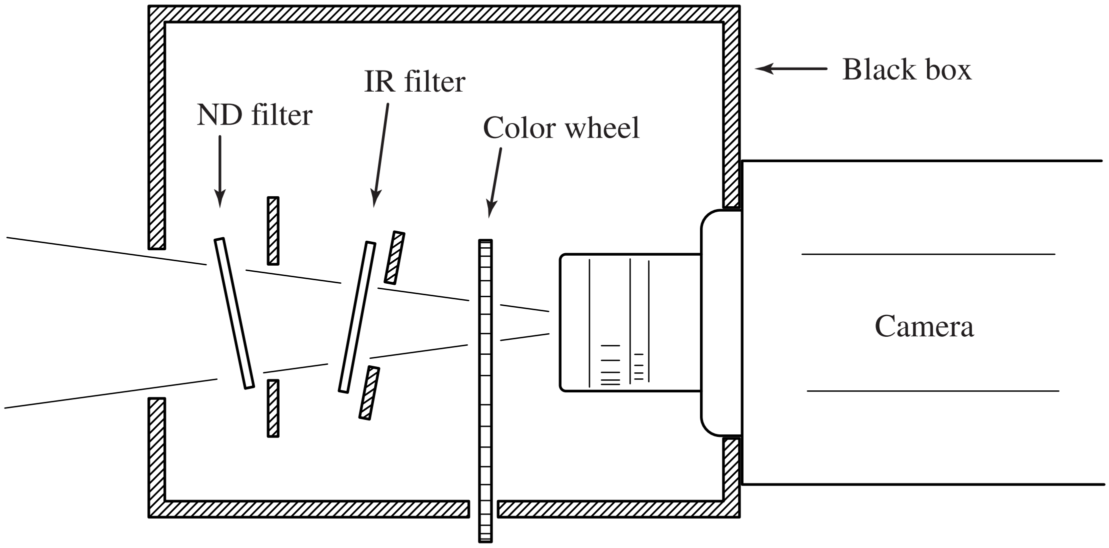


Figure 5.9: The configuration of filters used in front of the primary camera.

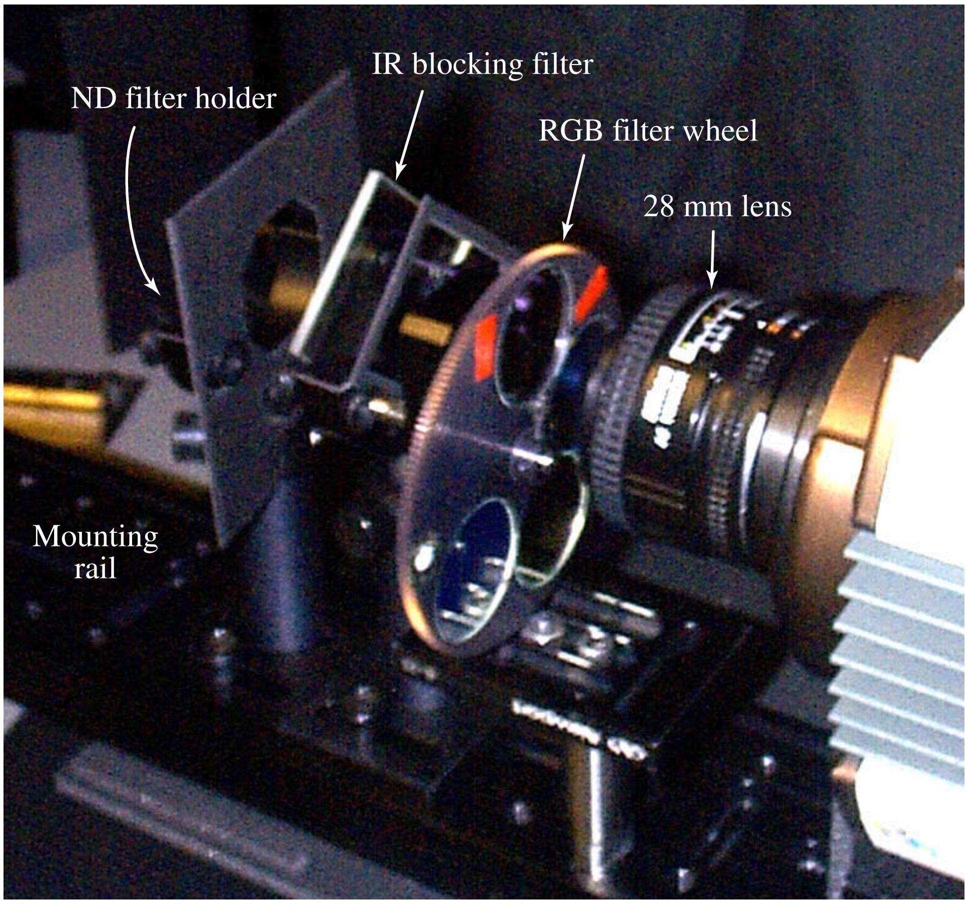


Figure 5.10: The experimental setup for image-based BRDF measurement.

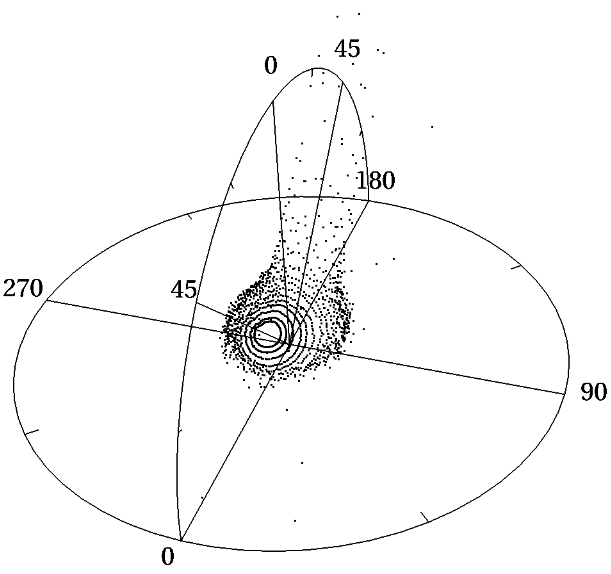


Figure 5.11: The BRDF samples generated by image-based measurement of the blue paint, corresponding to the 45° scattering diagram shown in Figure 5.24.

two colored paints, which had glossy finishes, with a gloss-reducing finish (Plasti-kote #33 "glass frosting spray"). We used the color separation filters when measuring the colored paints.

The data for each color channel of each measurement comprise approximately 1.5 million samples; the data corresponding to a single exitance angle for the blue channel of the blue paint are plotted in Figure 5.11.[5] We present the results of these measurements by showing 3D scattering diagrams produced using the reconstruction technique of Section 5.7. In these diagrams, the first parameter of f_r (either the incidence or exitance direction; by reciprocity they are equivalent) is held at a fixed value ω_0, and the distance from the origin to the displayed surface in the direction ω is equal to the reconstructed value of $f_r(\omega_0, \omega)$. Figure 5.21 shows the scattering diagrams of the primer for six incidence angles. Note how the surface is predominantly diffuse for normal incidence, but becomes strongly directional as the incidence angle increases. As expected, the primer shows some retroreflection, and its forward scattering lobe becomes larger and more off-specular as the angle of incidence increases.

Figures 5.22–5.24 and Figures 5.25–5.27 show the scattering diagrams for the three color channels of the red and blue paints for six different incidence angles. These diagrams reveal a distinct difference between the behaviors of these two paints. The blue paint exhibits

[5]These points are not a subset of the actual measurements; see Section 5.4.

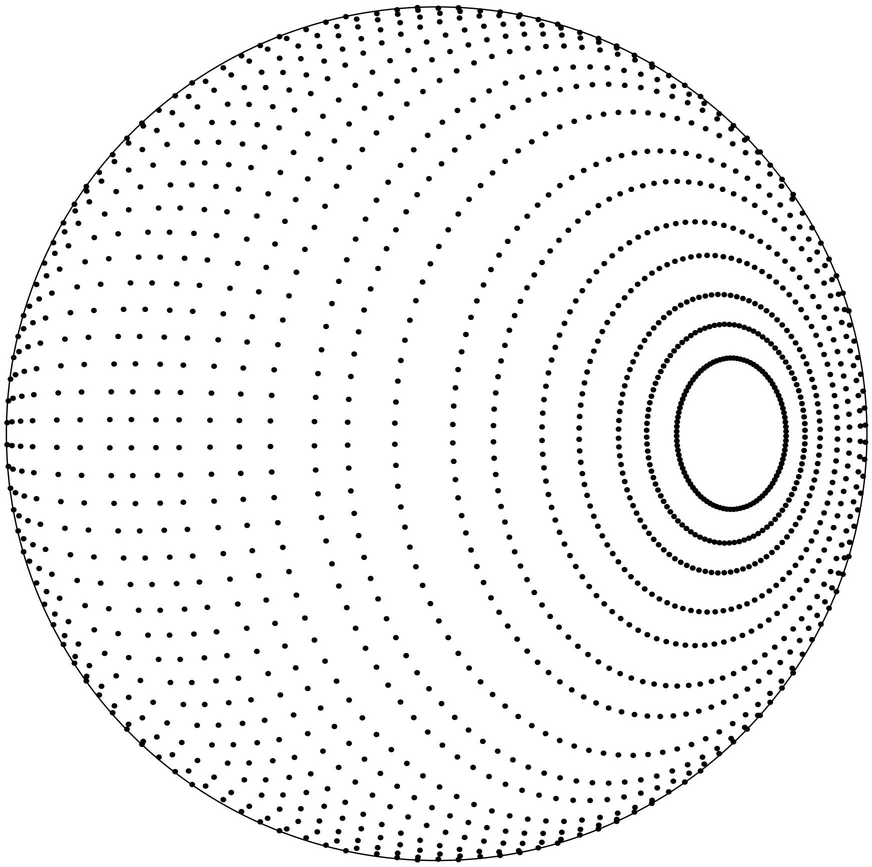


Figure 5.12: The sampling pattern from Figure 5.11 projected down to the disc. Each point is the projection onto the unit disc of 3D unit vector that represents ω_e for that sample point. Each circle corresponds to a single measurement image, and in this projection the shape of the rings is more evident than it is in Figure 5.11.

Figure 5.13: Photographs of the actual test samples used.

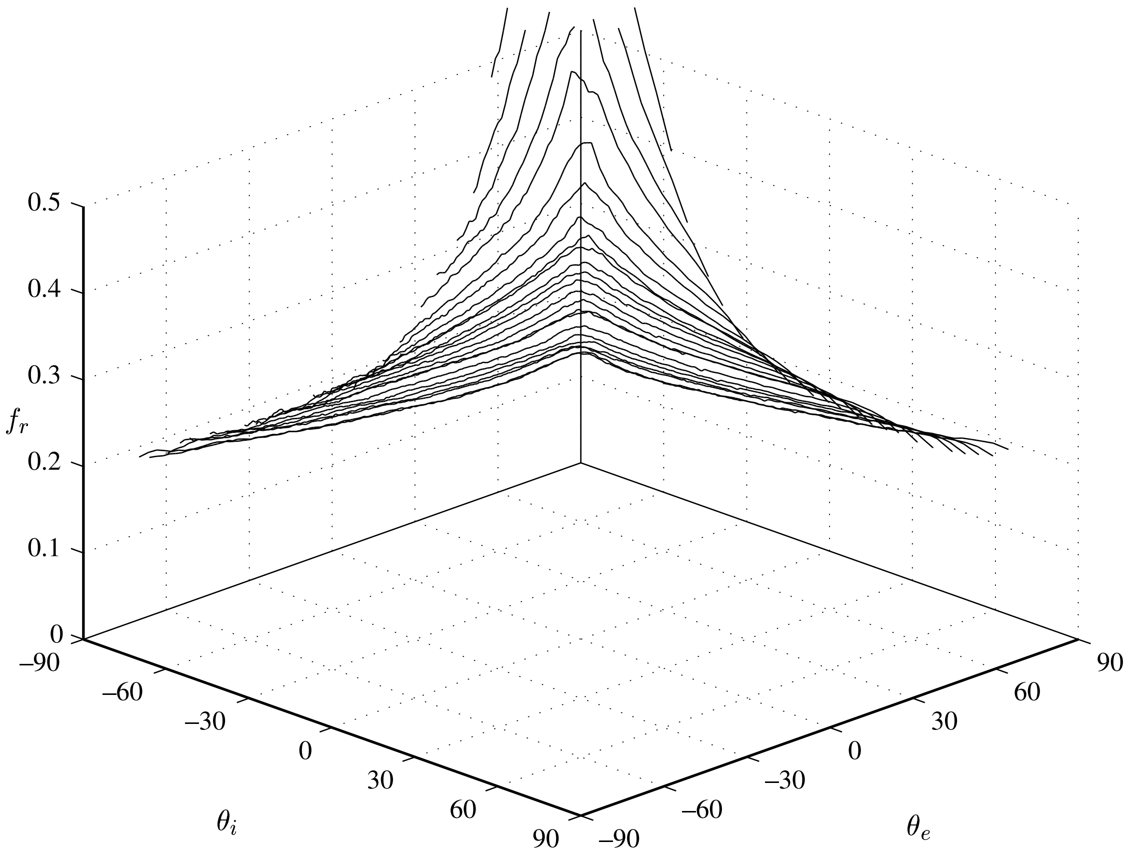


Figure 5.14: Incidence plane measurements of the BRDF of paper plotted against incidence and exitance angles, on a linear scale.

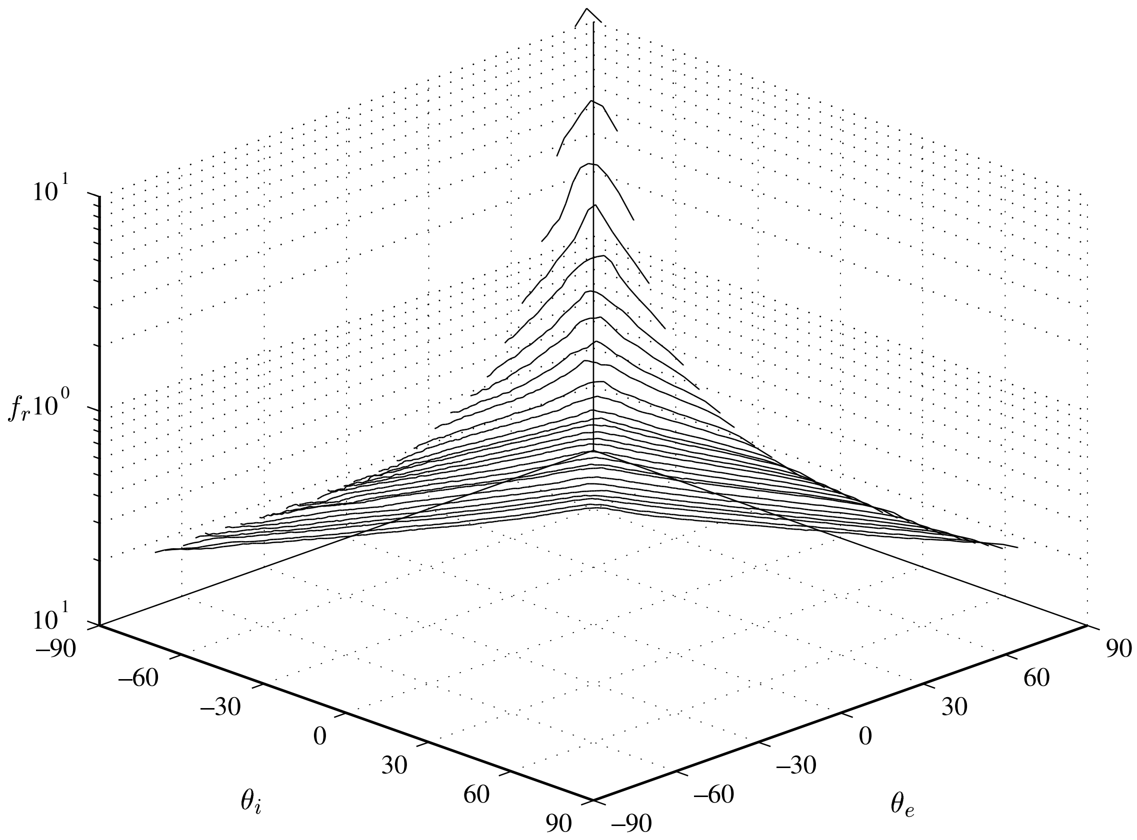


Figure 5.15: Incidence plane measurements of the BRDF of paper plotted against incidence and exitance angles, on a logarithmic scale.

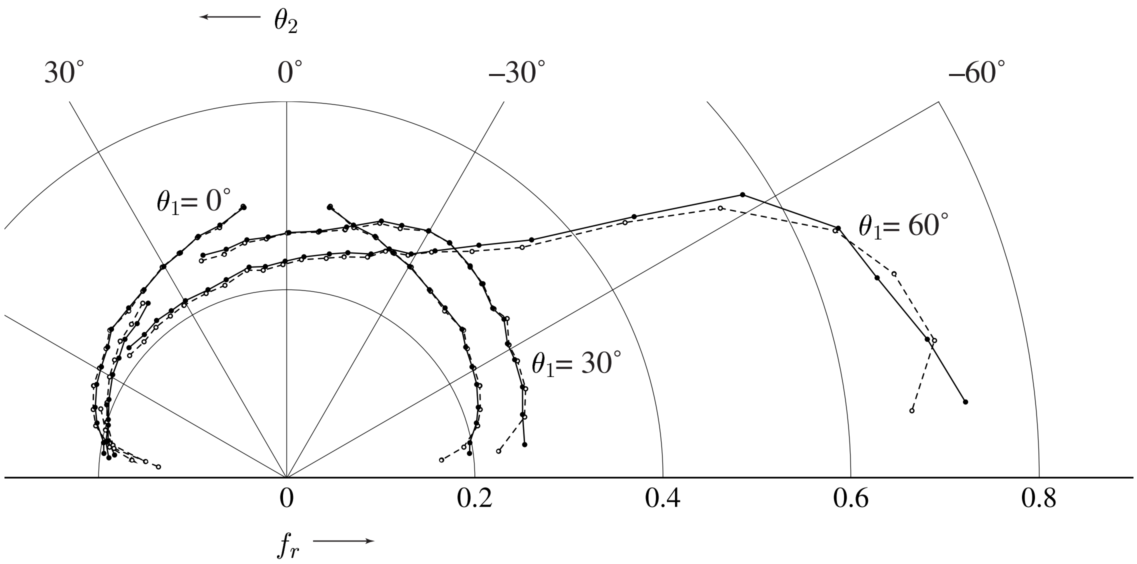


Figure 5.16: Reciprocity comparison. BRDF measurements of white paper are plotted for fixed exitance (solid lines) and fixed incidence (dashed lines) at $\theta = 0°$, $30°$, and $60°$.

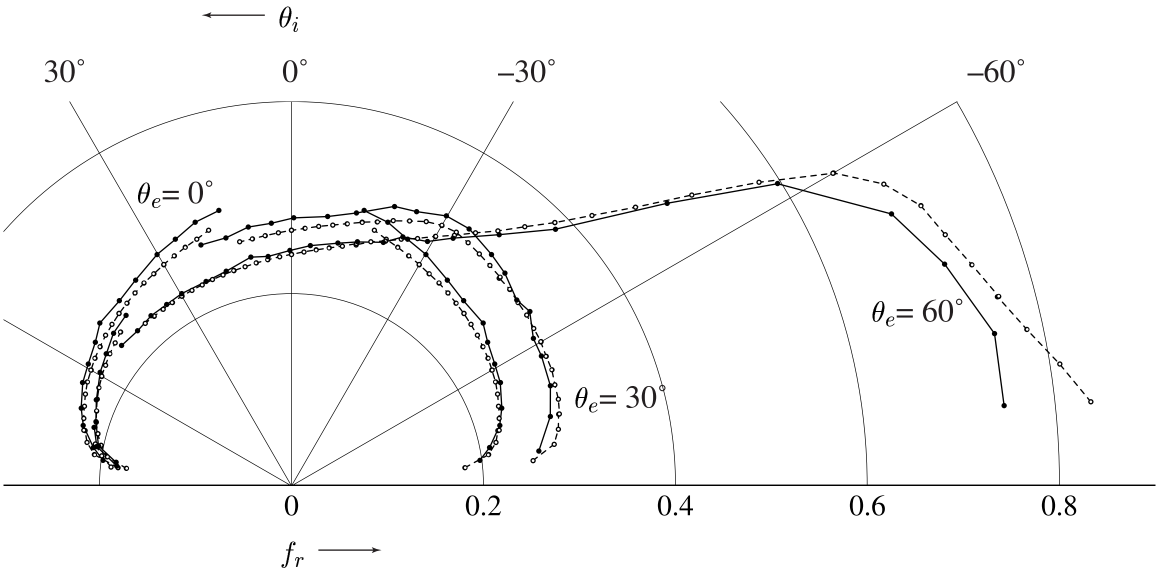


Figure 5.17: Gonioreflectometer comparison for office paper. The image-based BRDF measurements (solid lines) are plotted against gonioreflectometer measurements of the same material (dashed lines) at $\theta = 0°$, $30°$, and $60°$.

specularity that is mostly uncolored, as shown by the nearly constant size of its specular lobe. The blue color comes strictly from the diffuse component: comparing the scattering diagrams clearly shows a large diffuse component in the blue channel that becomes much smaller in the other two channels. The red paint, on the other hand, has almost no diffuse component. Its color comes from a broad directional peak, visible in the red channel, and its dimmer white highlight comes from a narrower, non-wavelength-dependent peak, visible in the green and blue channels. Both paints show a distinct toe, or increase in forward scattering when one angle nears grazing, which we conjecture may be due to scattering in the layer created by the gloss-reducing spray.

We measured the red and blue paints with the gonioreflectometer to validate our results over the entire hemisphere. Figures 5.28 and 5.29 show the gonioreflectometer measurements alongside the image-based measurements from Figures 5.22–5.27. The similarity of these plots indicates that our technique has successfully captured the BRDF.

5.6 Mapping the BRDF Domain to 3-space

Once we have measured a surface, we have a large collection of BRDF samples, each with a different configuration of illumination and viewing directions. Each such configuration is a point in the BRDF's domain, and we can think of our measurements as being scattered through a three-dimensional space. Furthermore, as we observed at the start of this chapter,

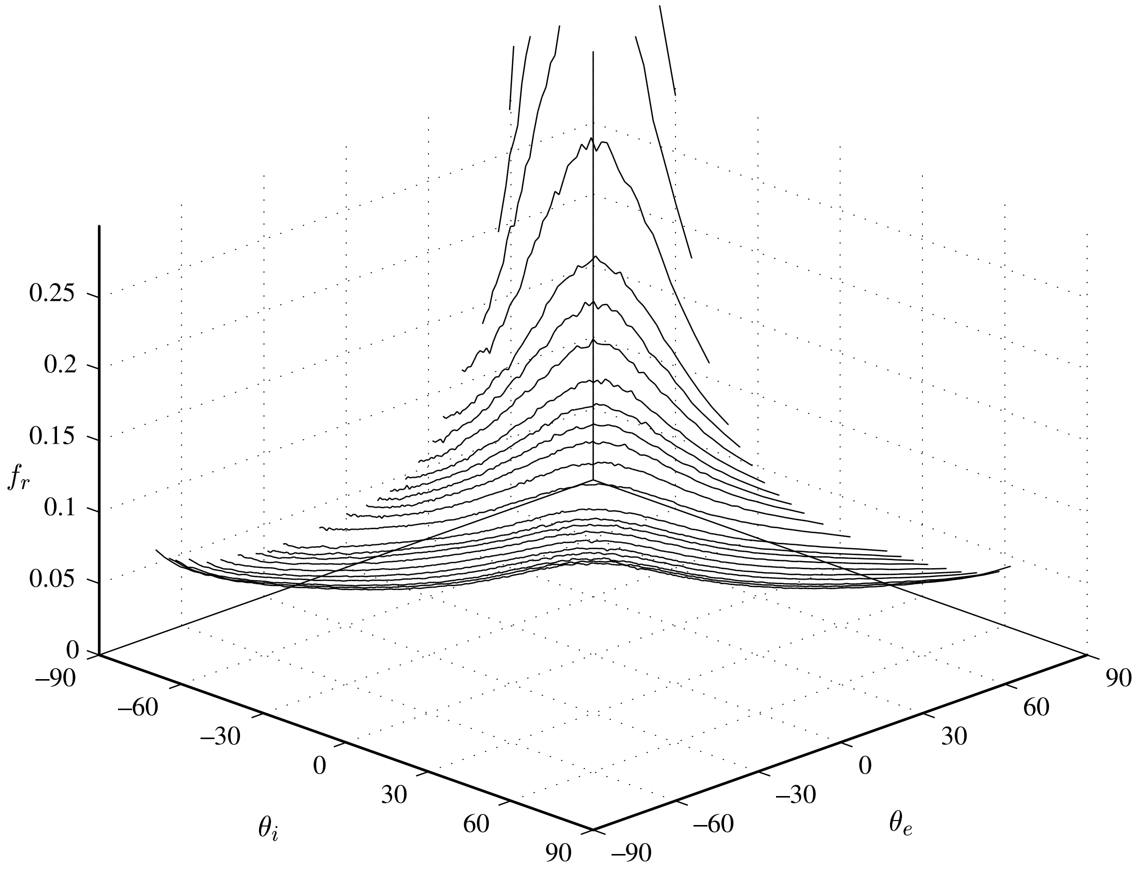


Figure 5.18: Incidence plane BRDF measurements for the gray primer plotted against incidence and exitance angles, on a linear scale.

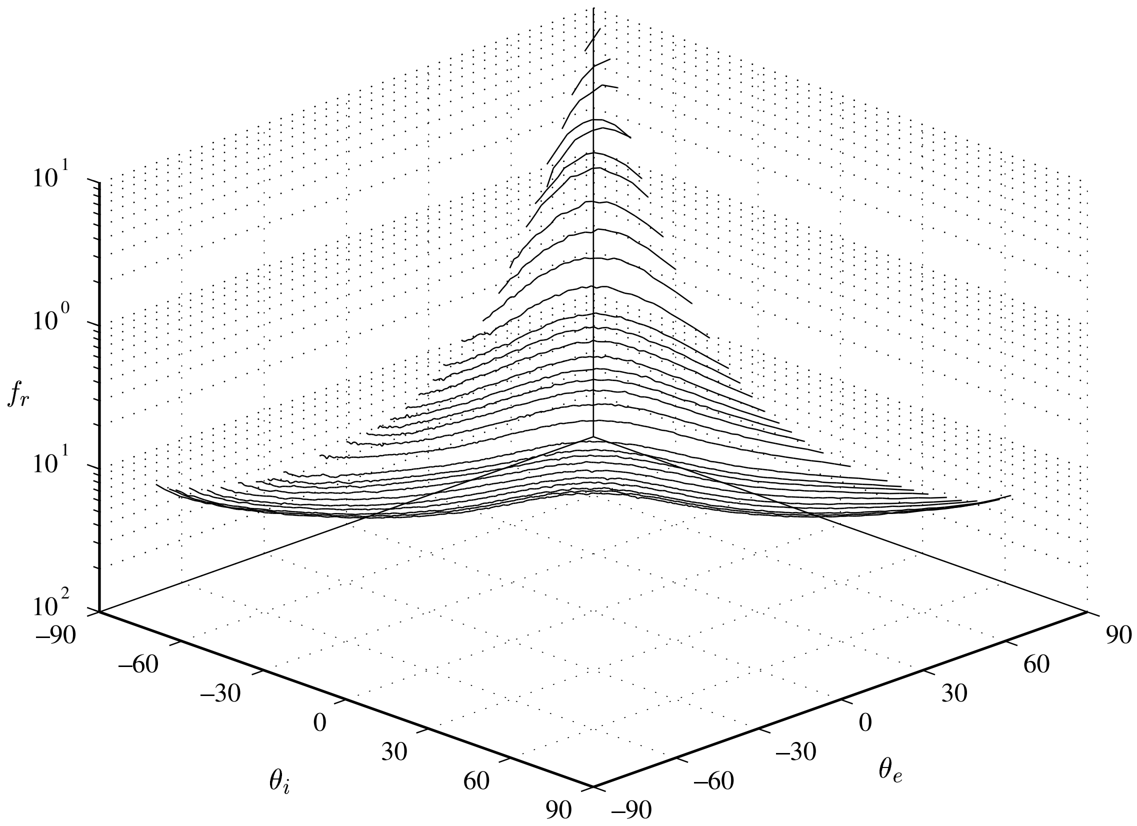
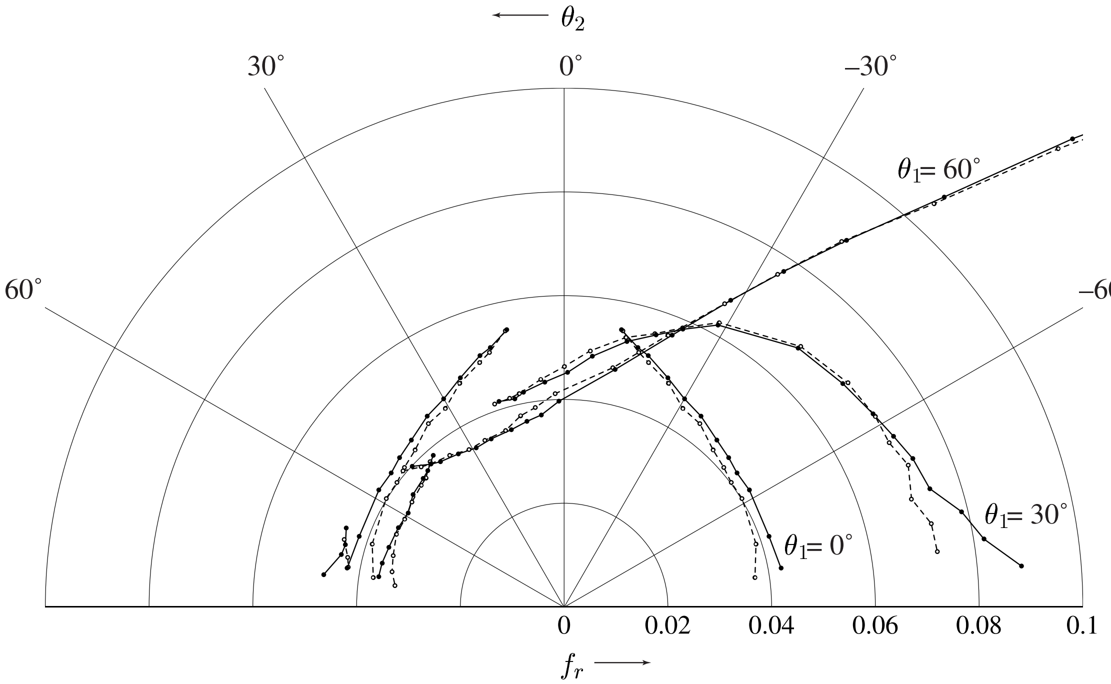


Figure 5.19: Incidence plane BRDF measurements for the gray primer plotted against incidence and exitance angles, on a logarithmic scale.



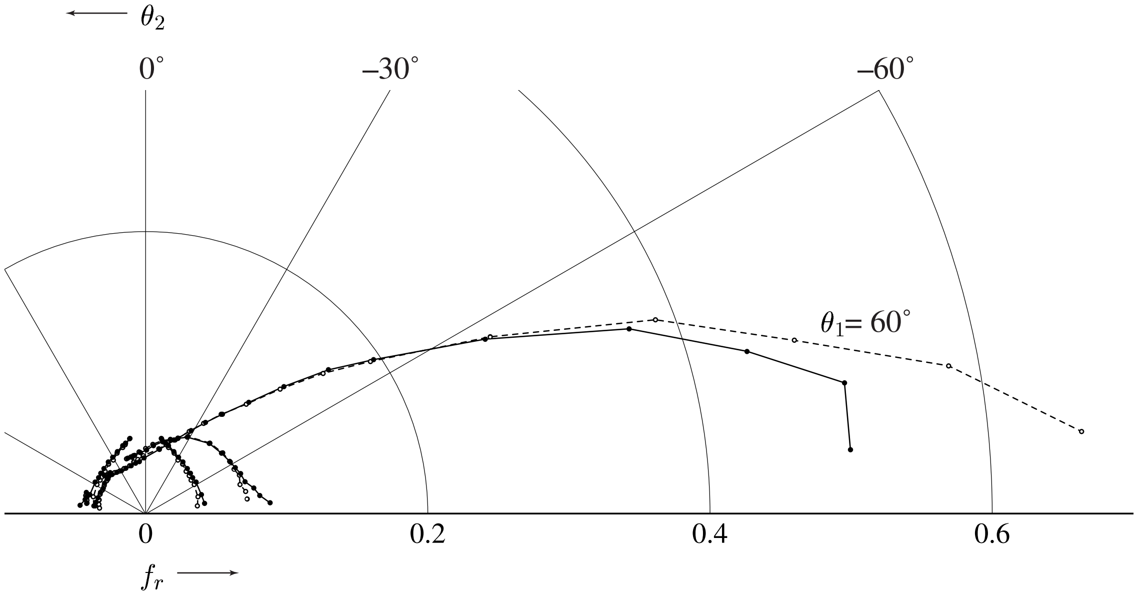


Figure 5.20: Reciprocity comparison for gray primer. BRDF measurements of the primer are plotted for fixed exitance (solid lines) and fixed incidence (dashed lines) at $\theta = 0°$, $30°$, and $60°$. The plot is shown at two scales so that all three curves can be seen clearly.

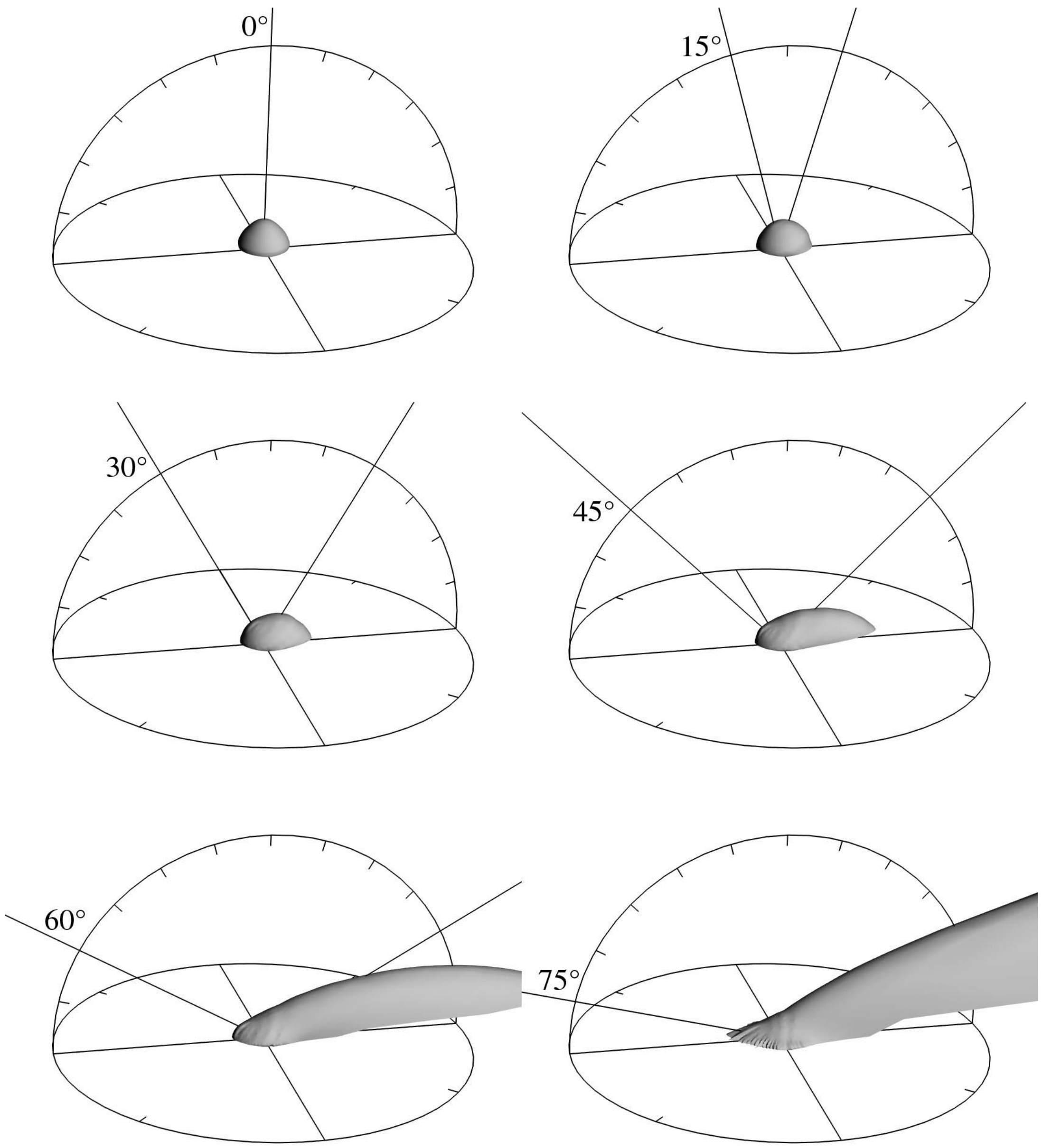


Figure 5.21: Resampled scattering diagrams of the BRDF measurements of the gray primer at various incidence angles. The outer ring corresponds to a BRDF value of $1/\pi$.

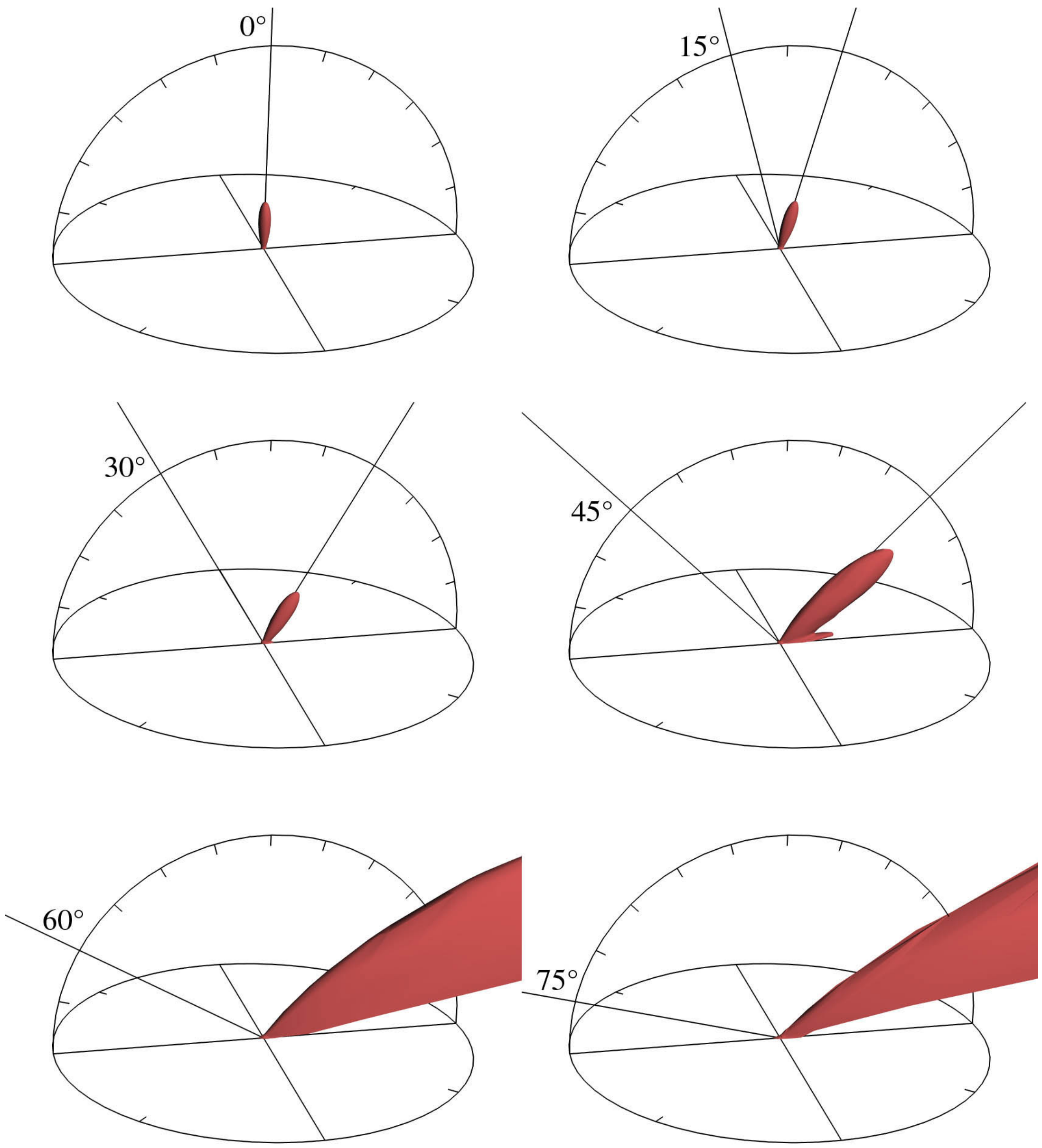


Figure 5.22: Resampled scattering diagrams of the BRDF measurements of the blue enamel paint through the red filter. The outer ring corresponds to a BRDF value of $1/2\pi$.

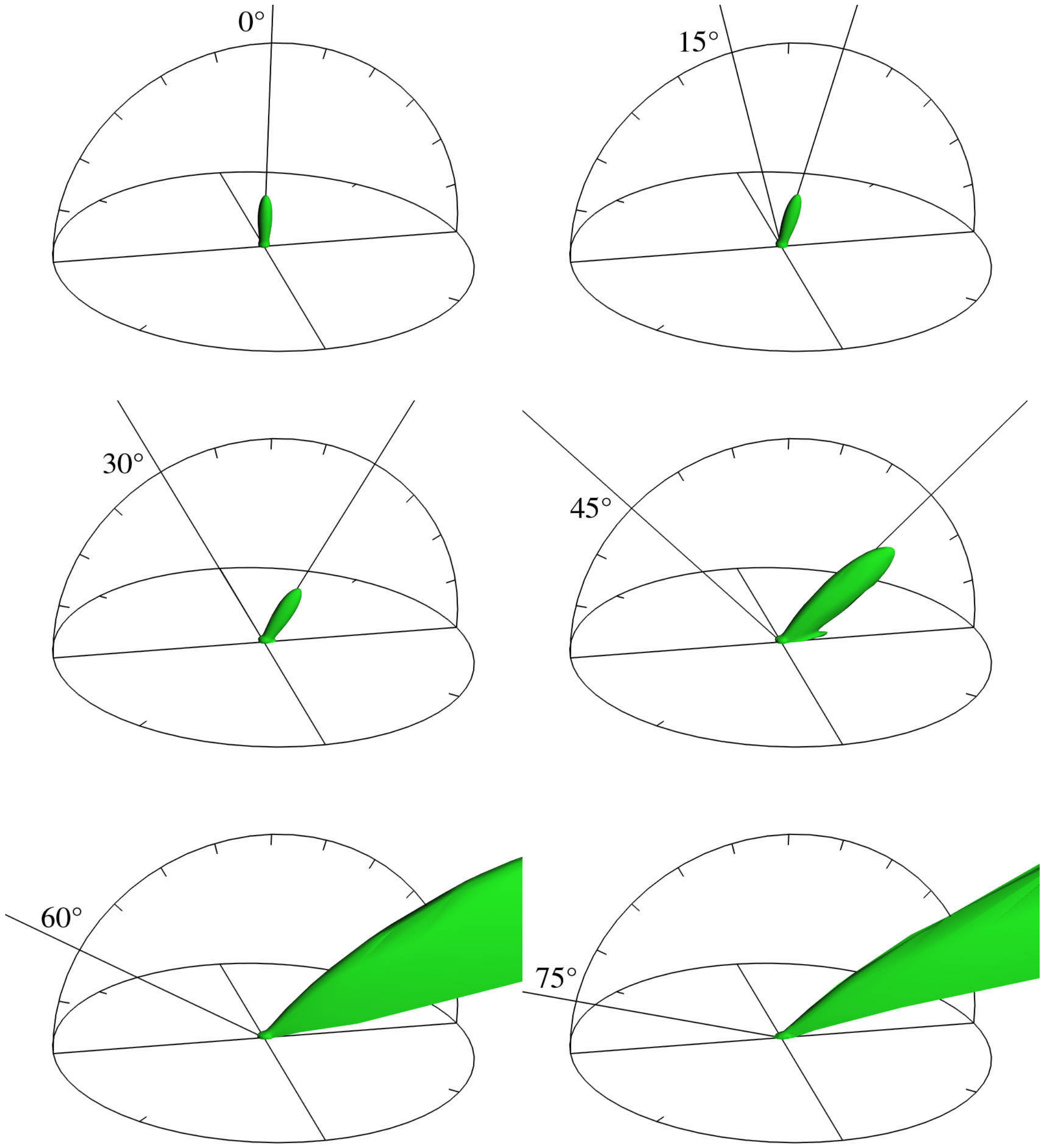


Figure 5.23: Resampled scattering diagrams of the BRDF measurements of the blue enamel paint through the green filter. The outer ring corresponds to a BRDF value of $1/2\pi$.

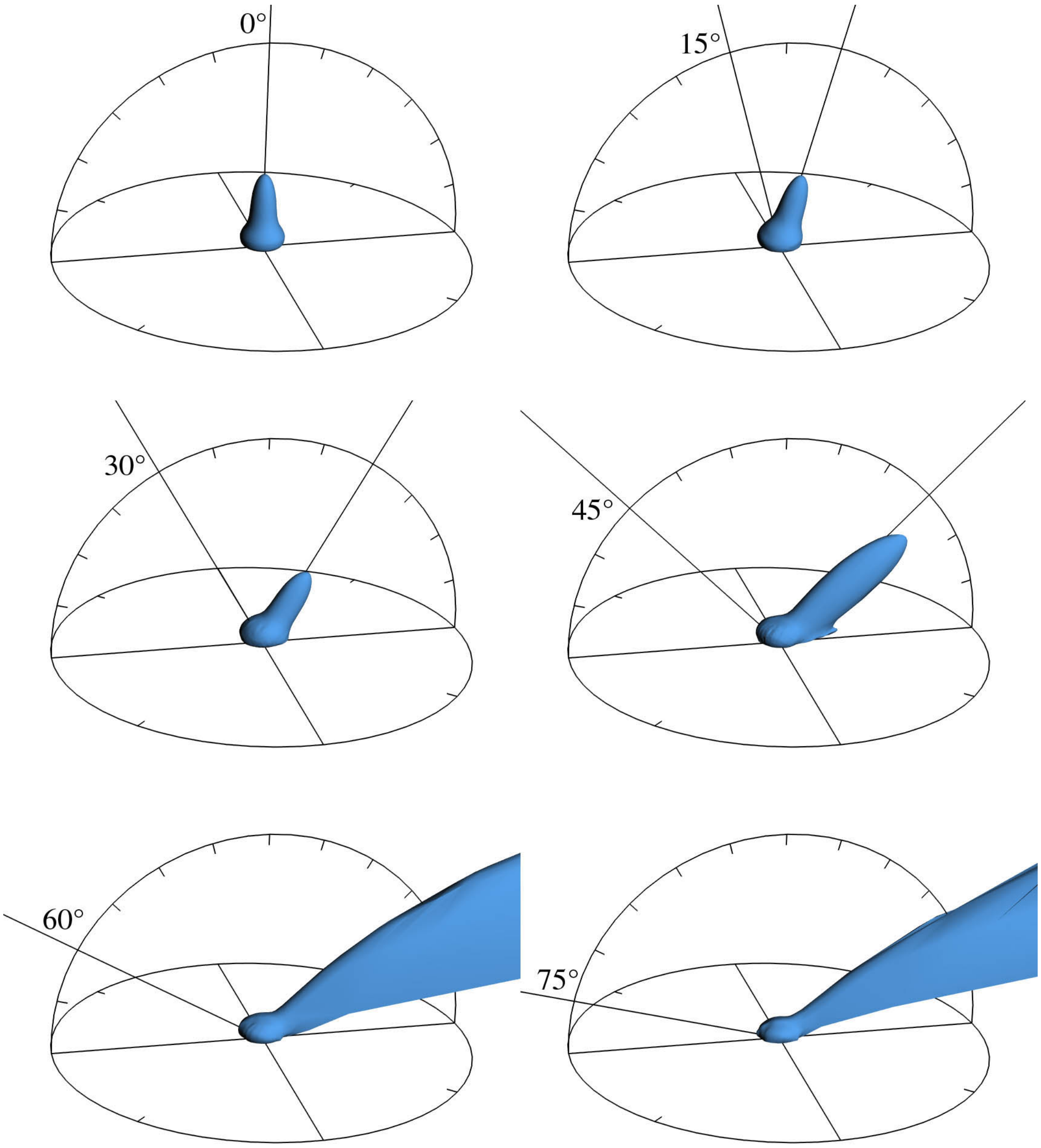


Figure 5.24: Resampled scattering diagrams of the BRDF measurements of the blue enamel paint through the blue filter. The outer ring corresponds to a BRDF value of $1/2\pi$.

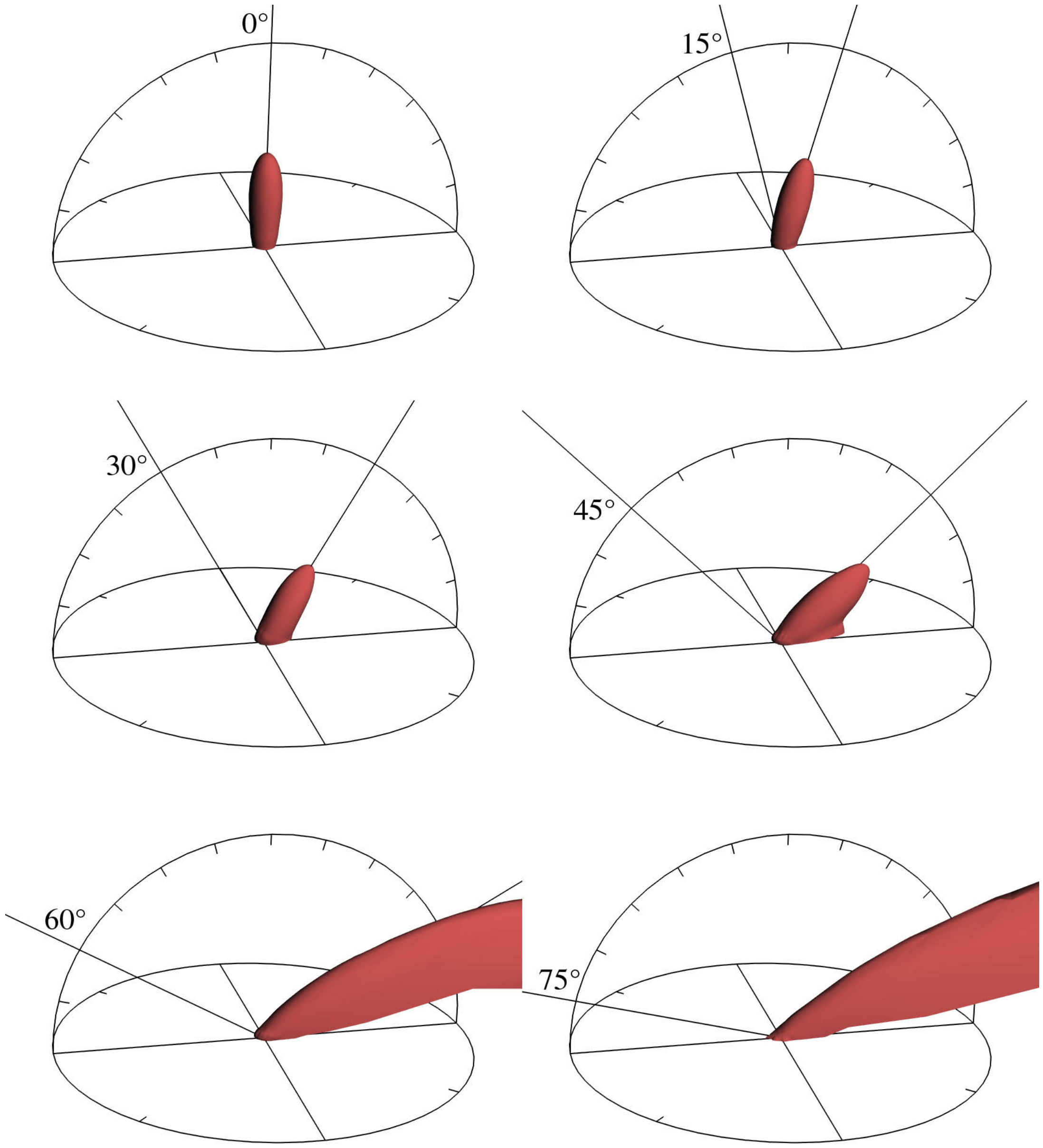


Figure 5.25: Resampled scattering diagrams of the BRDF measurements of the red metallic lacquer paint through the red filter. The outer ring corresponds to a BRDF value of $1/2\pi$.

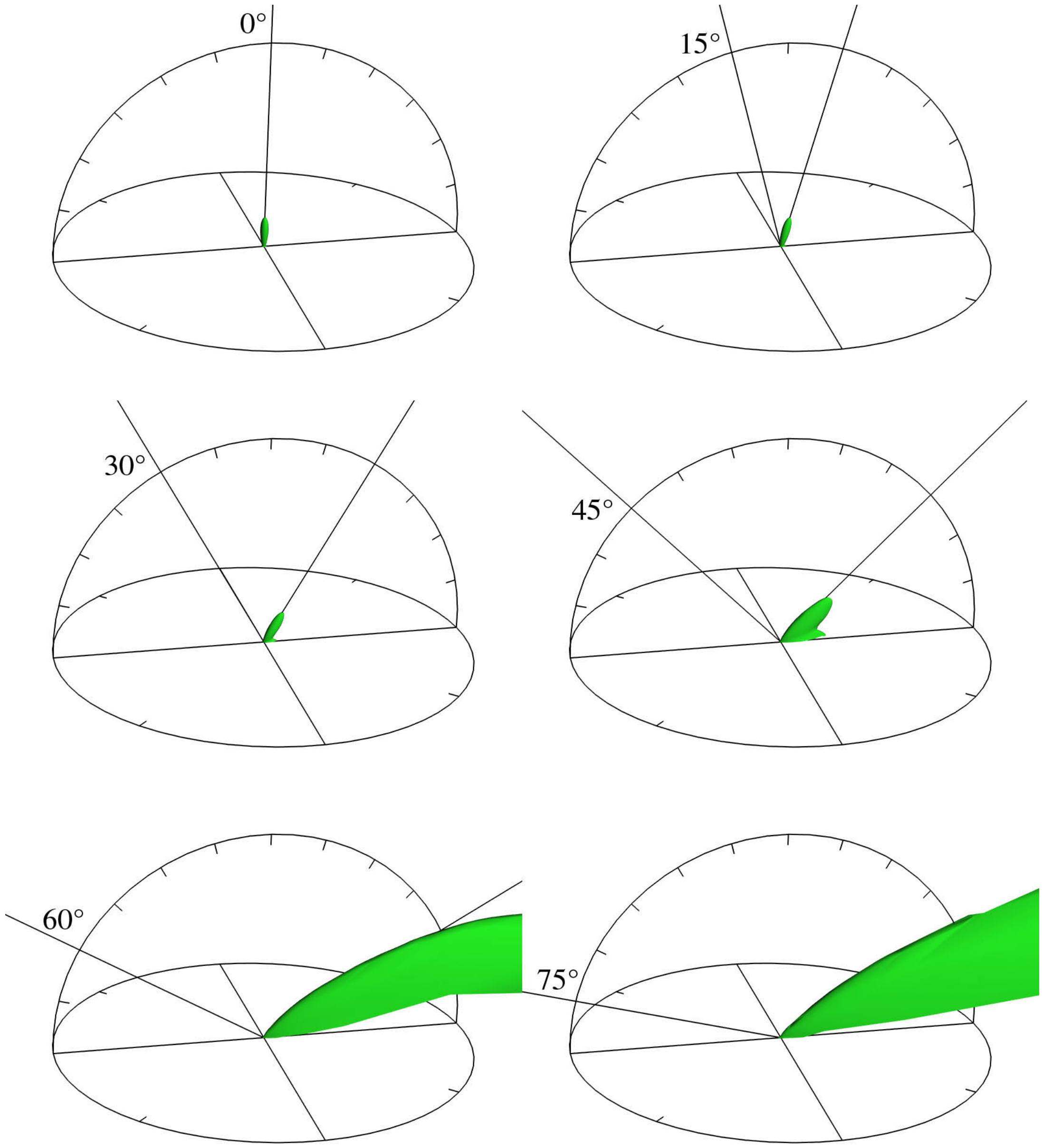


Figure 5.26: Resampled scattering diagrams of the BRDF measurements of the red metallic lacquer paint through the green filter. The outer ring corresponds to a BRDF value of $1/2\pi$.

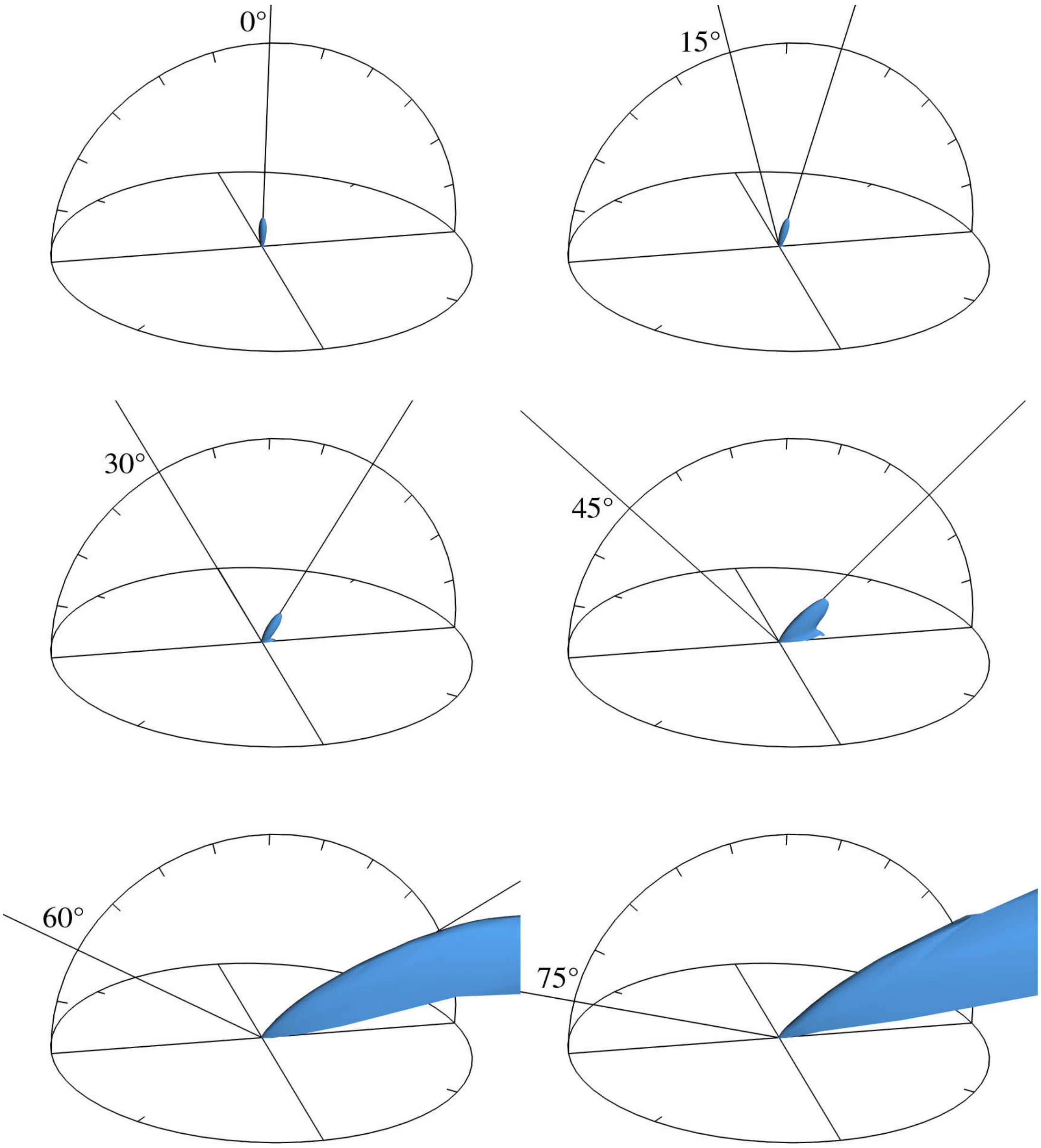


Figure 5.27: Resampled scattering diagrams of the BRDF measurements of the red metallic lacquer paint through the blue filter. The outer ring corresponds to a BRDF value of $1/2\pi$.

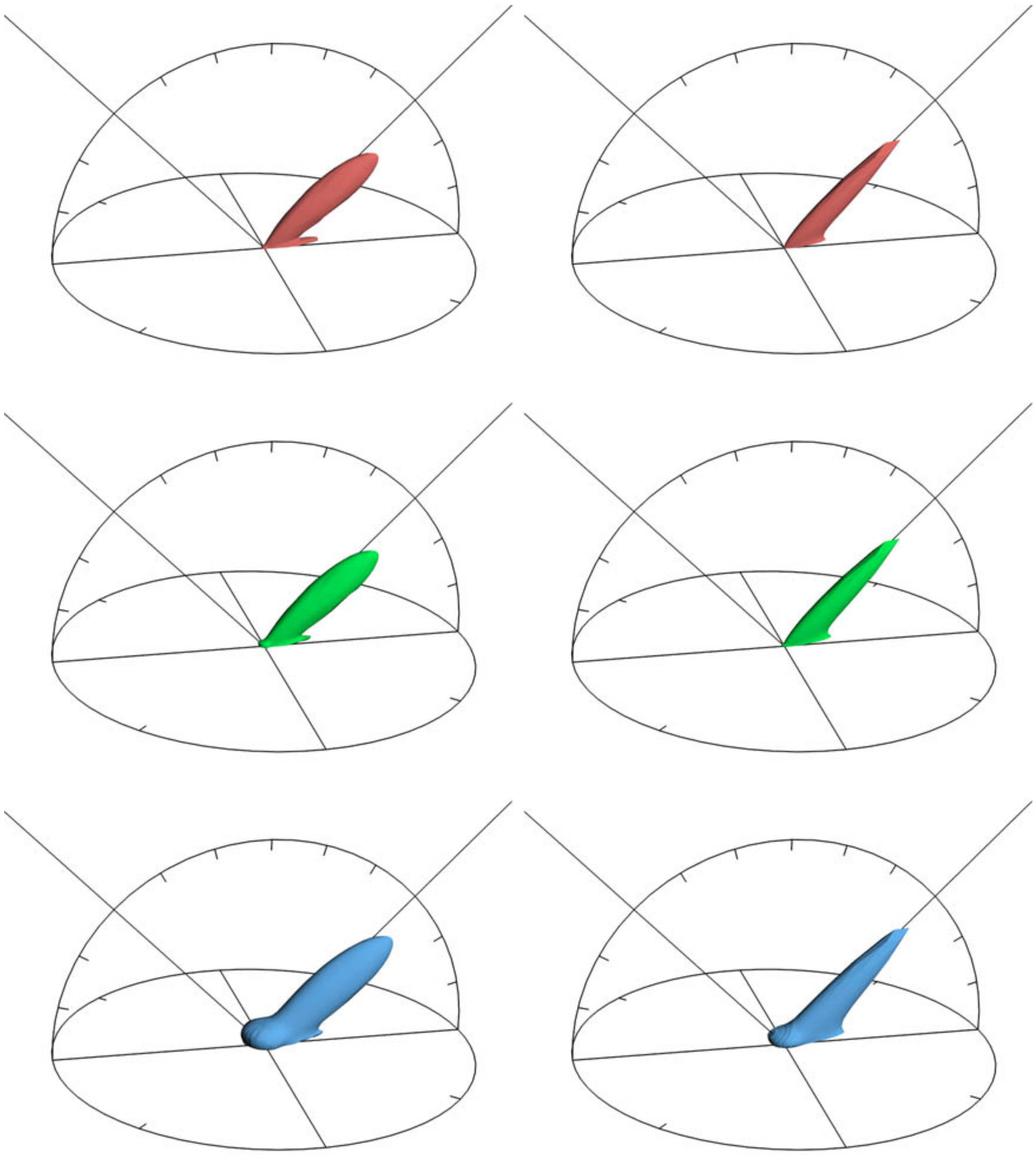


Figure 5.28: The image-based measurements of the blue paint for an angle of incidence of 45° (at left; note that these plots are repeated from Figures 5.22–5.24) with the corresponding measurements from the gonioreflectometer (at right). The gonioreflectometer data have been triangulated directly from the sample points, while the image-based data, which do not come in sets of fixed θ, have been resampled.

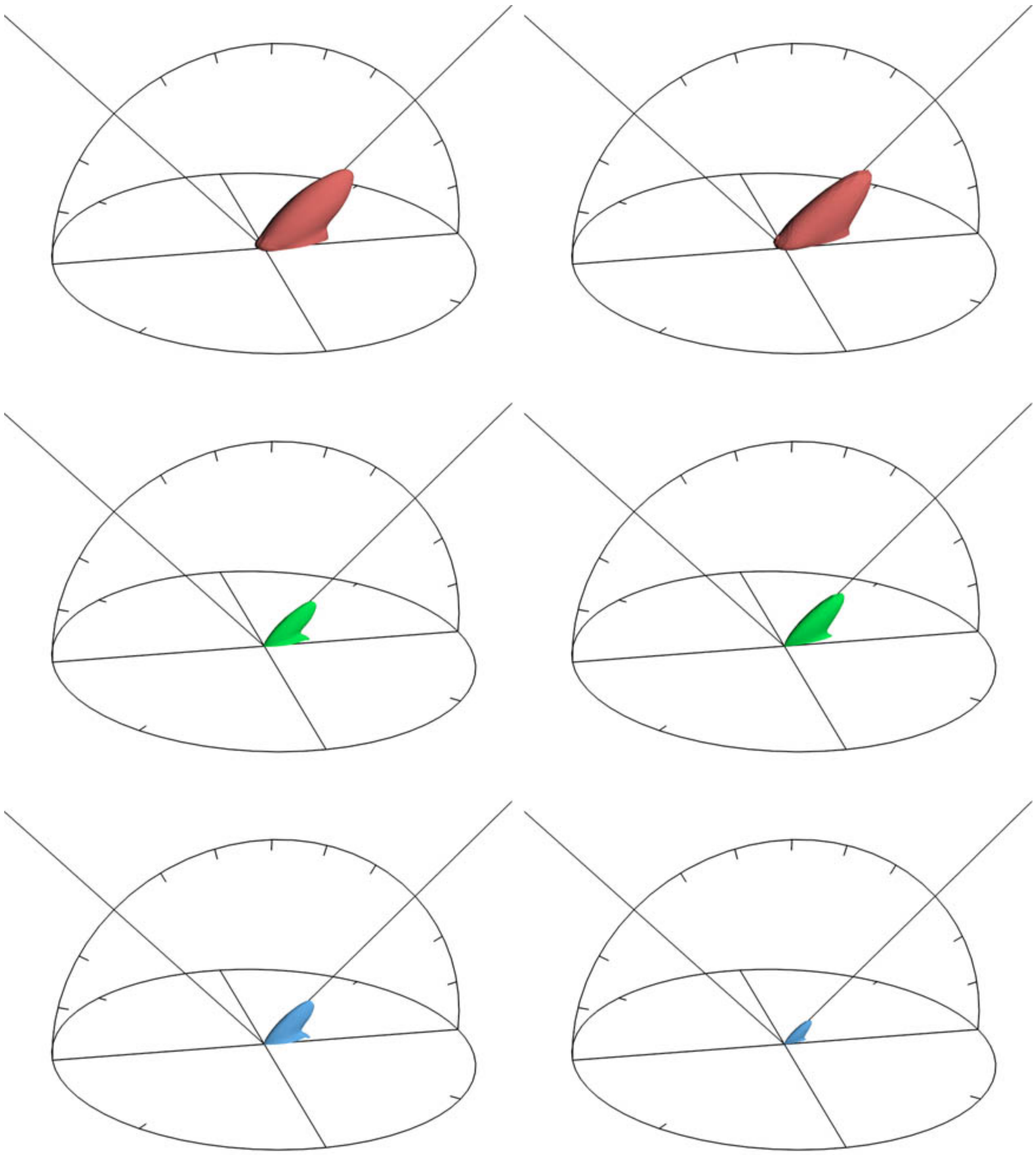


Figure 5.29: The image-based measurements of the red paint for an angle of incidence of 45° (at left; note that these plots are repeated from Figures 5.25–5.27) with the corresponding measurements from the gonioreflectometer (at right). The gonioreflectometer data have been triangulated directly from the sample points, while the image-based data, which do not come in sets of fixed θ, have been resampled.

these points are arranged on densely sampled sheets. In order to conveniently represent the sample positions, either to visualize them using 3D rendering or to perform resampling computations, we would like to have a coordinate system that organizes the domain of an isotropic BRDF in a volume of ordinary 3D space in a way that is easily understood, for visualization, or computationally efficient, for resampling.

To state the problem in more specific terms, we seek a function $\psi : \Omega \times \Omega \to \mathbb{R}^3$ with the following properties:

1. *Continuity.* To avoid introducing artifacts into images computed using our coordinates, ψ must be C^2 continuous.

2. *Bijectivity.* Two configurations should map to the same point if and only if they are required to have the same BRDF value. In particular, rotating a configuration about the normal or interchanging the two directions should leave the 3D coordinate unchanged.[6] Bilateral symmetry could also be included in this property.

   For visualization, it is acceptable or even desirable for reciprocal configurations to be considered different, since it may aid in understanding some aspects of the sampling pattern. However, reciprocity should map to an obvious symmetry; for example, reciprocal points might map to points that are symmetric across a plane.

In the following sections, we will consider three possibilities for ψ: one very simple but flawed, one useful for visualization, and one especially suited to resampling the data from our particular BRDF measurement system. For all three mappings, it will be convenient to use cylindrical coordinates to represent $\mathbb{R}^3$; we will adopt the convention that $(r, \phi, z)_c$ corresponds to the Cartesian point $(r \cos \phi, r \sin \phi, z)$ (Figure 5.30). Throughout this section (θ_i, ϕ_i) will be the spherical coordinates of ω_i and (θ_e, ϕ_e) will be the spherical coordinates of ω_e.

5.6.1 A simple cylindrical mapping

If we account for isotropy by keeping only the difference of ϕ_i and ϕ_e and map directly to cylindrical coordinates, we have the following candidate for ψ:

$$\psi_1(\omega_i, \omega_e) = (2\theta_i/\pi, \phi_e - \phi_i, 2\theta_e/\pi)_c. \tag{5.1}$$

Here are some characteristics of ψ_1:

- The set of all configurations that share a particular value of θ_e (an *incident hemisphere*) maps to a unit disc parallel to the x-y plane. Normal-exitance configurations map to the plane $z = 0$; grazing-exitance configurations map to $z = 1$.

[6] By using the term *bijective*, we are really calling ψ a function on $(\Omega \times \Omega) \,/ \sim_i$ or $(\Omega \times \Omega) \,/ (\sim_r \cup \sim_i)$, rather than $\Omega \times \Omega$.

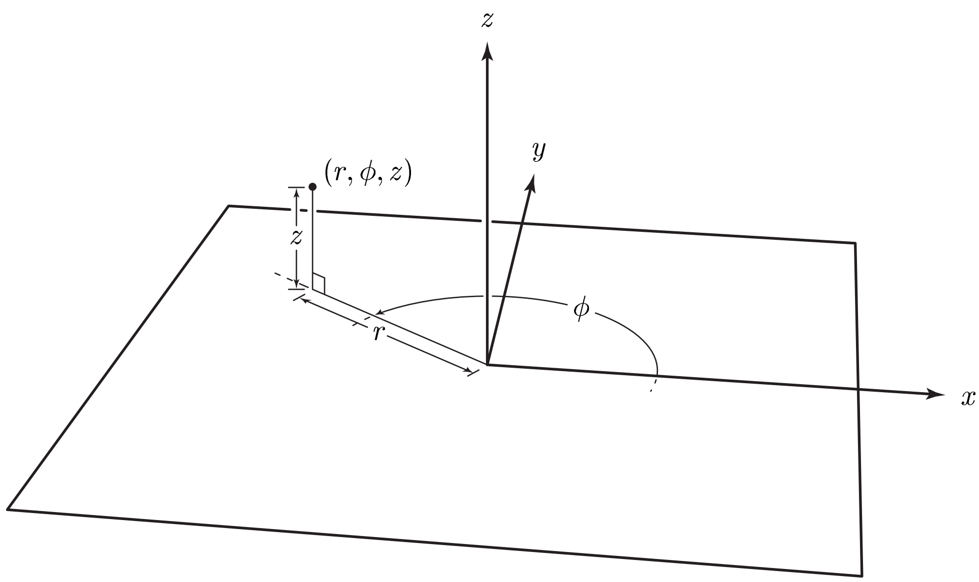


Figure 5.30: Cylindrical coordinates.

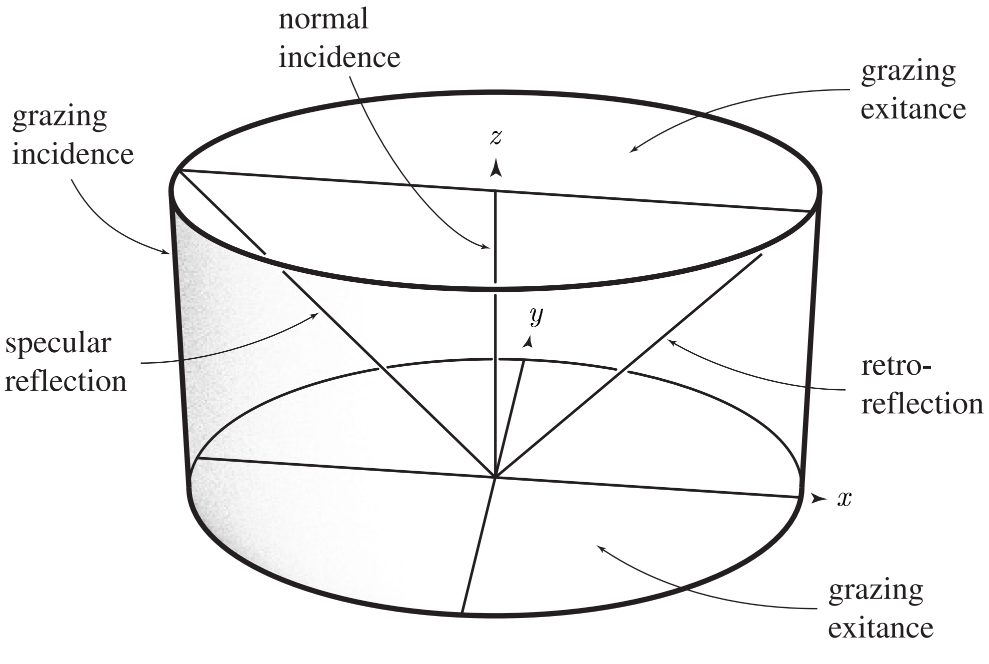


Figure 5.31: The cylinder corresponding to the mapping ψ_1.

- An exitant hemisphere maps to a cylinder centered on the z axis. Since a cylinder is topologically unlike a hemisphere, this is an indication that ψ_1 lacks some of the desired properties. Configurations with normal incidence map to the z axis, and grazing-incidence configurations map to the cylinder $r = 1$.
- The set of configurations for which $\theta_i = \theta_e$ maps to a $90°$ unit cone with its apex at the origin and its base on the plane $z = 1$. The ideal specular configurations, for which $\theta_i = \theta_e$ and $\Delta\phi = 180°$, are along the intersection of that cone with the plane $y = 0$, which is a line segment from the origin to the point $(-1, 0, 1)$. The retroreflective configurations, for which $\theta_i = \theta_e$ and $\Delta\phi = 0$, are along a similar segment from the origin to $(1, 0, 1)$ (Figure 5.31).

How does the symmetry of reciprocity appear after it has undergone this mapping? The relationship $(\omega_i, \omega_e) \leftrightarrow (\omega_e, \omega_i)$ becomes $(r, \phi, z)_c \leftrightarrow (z, -\phi, r)_c$. On the cone $r = z$, this is simply a reflection across the plane $y = 0$; and if we restrict our attention to the square for which $\Delta\phi = 0$, the reflection is across the line $z = r$. The normal-exitance set, the disc at $z = 0$, is symmetric with the normal-incidence set, the z axis; and the grazing-exitance set, the disc at $z = 1$, is symmetric with the grazing-incidence set, the cylinder at $r = 1$. The symmetry on the entire space may be described as a reflection across the cone $z = r$ followed by a reflection across the plane $y = 0$. If we consider both bilateral symmetry and reciprocity, we can think of these two symmetries independently: a simple bilateral one and a reflection through a cone.

The mapping ψ_1 fails in both bijectivity and continuity. It is not bijective because the some sets of configurations that are a single point as far as isotropic BRDF is concerned do not map to single points. For instance, a set of configurations for which $\theta_i = 0$ and θ_e is fixed can have only a single BRDF value, since all such configurations are equivalent under rotation about the surface normal. However, ψ_1 maps such sets to circles on the disc $z = 0$. It is also discontinuous, because points near $\theta_i = 0$ are mapped to far-apart points around that circle. This is despite the fact that ψ_1 is obviously a continuous function of θ_i, θ_e, ϕ_i, and ϕ_e. The mapping from the hemisphere to spherical coordinates is what is discontinuous.

5.6.2 A mapping for visualization

The continuity and bijectivity faults of ψ_1 can be remedied by the following mapping:

$$\psi_2(\omega_i, \omega_e) = (\sin\theta_i \sin\theta_e, \Delta\phi, \cos\theta_i - \cos\theta_e)_c. \tag{5.2}$$

Under this mapping, the images of normal-incidence and -exitance configurations do not depend on $\Delta\phi$, as they should not under anisotropy. This eliminates the problems with continuity and bijectivity that we saw with ψ_1.

The geometric relationship between the ranges of ψ_1 and ψ_2 is illustrated in Figure 5.32. Reciprocity and bilateral symmetry in this case are reflections across the x-y and y-z planes. The loci of normal incidence, normal exitance, specular reflection, and retroreflection form a cross in the y-z plane (Figure 5.33). The grazing configurations are at the surface of the sphere.

The loci of configurations for fixed incidence or exitance angles are no longer discs under ψ_2; they are hemi-ellipsoids. For instance, look at the set of configurations for which $\theta_i = \theta_i^0$; the image of any point in this set has the form $(a \sin\theta, \phi, b + \cos\theta)$ for the fixed values $a = \sin\theta_i^0$ and $b = -\cos\theta_i^0$. These points may easily be recognized as lying on an ellipsoid with principal radii of 1, $\sin\theta_i^0$, and $\sin\theta_i^0$ and center at $(0, 0, -\cos\theta_i^0)$. In fact, since $\theta > 0$, they all lie on the upper hemi-ellipsoid. This hemi-ellipsoid is a hemisphere when $\theta_i^0 = 90°$, and as θ_i^0 approaches zero it becomes increasingly narrow until it collapses to a line at $\theta_i^0 = 0$.

Visualizing configurations as points within the spherical volume defined by ψ_2 allows configurations to be identified as having one or both angles near normal or grazing, displays nearness to specularity or retroreflection, and allows bilaterally symmetric or reciprocal configurations to be identified easily.

5.6.3 A mapping for resampling

The function ψ_2 has desirable properties for visualization, but the measurements from our image-based BRDF measurement technique come on sheets with near-constant $\langle\omega_i, \omega_e\rangle$, which do not map to any particularly simple surfaces under ψ_2. Since the density of samples is drastically different along and across the sheets, it is desirable to be able to filter the points with an anisotropic kernel,[7] an operation that can be made vastly more efficient if the sampling sheets coincide with the coordinate planes. Also, it is desirable to be able to reconstruct a BRDF using reciprocal points interchangeably to in effect double the sampling density. To this end, we present a third mapping, ψ_3:

$$\psi_3(\omega_i, \omega_e) = (\sin\theta_i \sin\theta_e, \Delta\phi, \cos\theta_i \cos\theta_e)_c. \tag{5.3}$$

This mapping differs from ψ_2 only in the z coordinate, but it has the remarkable property that loci of configurations with constant $\langle\omega_i, \omega_e\rangle$ become parallel planes, which can easily be rotated to coincide with the coordinate planes. To see that this is true, consider the unit vectors ω_i and ω_e in Cartesian coordinates. Assuming without loss of generality that $\phi_i = 0$, the coordinates of the two vectors are $(\sin\theta_i, 0, \cos\theta_i)$ and $(\cos\Delta\phi \sin\theta_e, \sin\Delta\phi \sin\theta_e, \cos\theta_e)$. The inner product of these vectors is $\cos\Delta\phi \sin\theta_i \sin\theta_e + \cos\theta_i \cos\theta_e$, which is the sum of

[7]The anisotropy of reconstruction kernels should not be confused with the anisotropy of BRDFs.

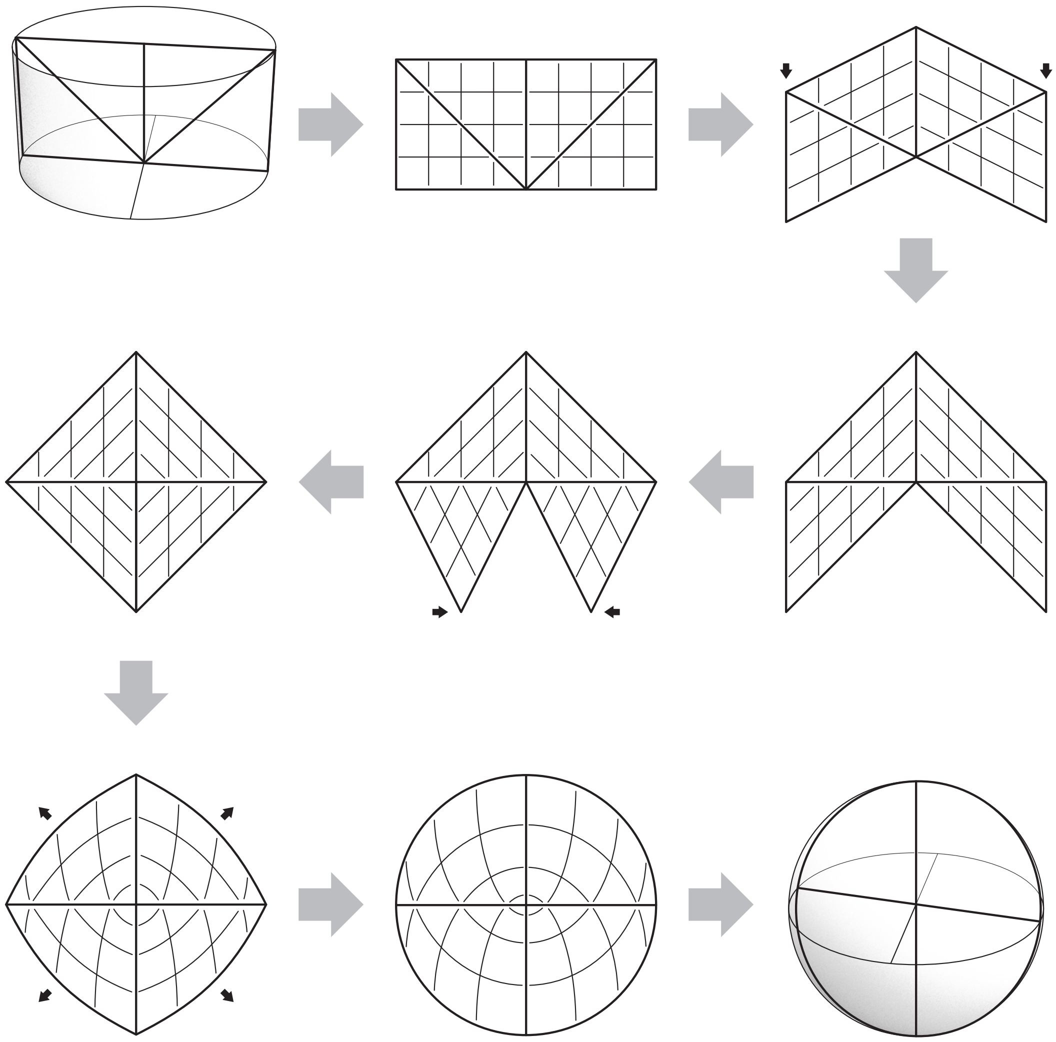


Figure 5.32: The relationship between ψ_1 and ψ_2. The first and last illustrations correspond to Figures 5.31 and 5.33; the intermediate steps show how a planar slice of the cylinder can be deformed, in the plane, into a corresponding planar slice of the sphere.

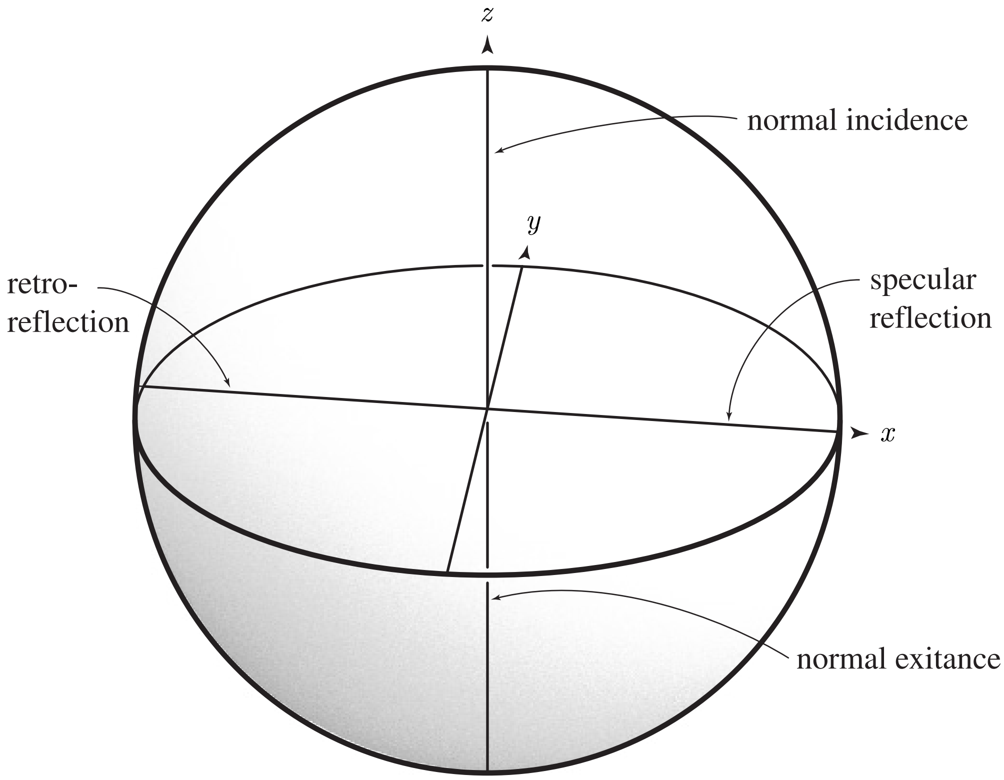


Figure 5.33: The sphere corresponding to the mapping ψ_2.

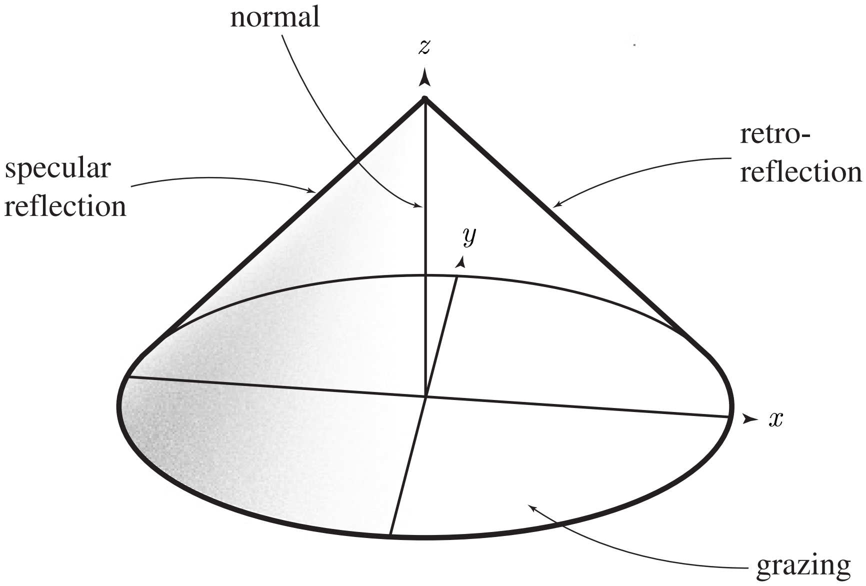


Figure 5.34: The cone corresponding to the mapping ψ_3.

the y and z coordinates of $\psi_3(\omega_i, \omega_e)$. Thus if $\langle \omega_i, \omega_e \rangle = c$ then (ω_i, ω_e) maps under ψ_3 to a point on the plane $y + z = c$.

Figure 5.34 indicates the images of several interesting sets under ψ_3. The image of the entire domain is a cone. It is obvious from Equation 5.3 that ψ_3 is invariant with respect to exchanging ω_i and ω_e, so reciprocal configurations map to the same point. Again, the normal-incidence or -exitance configurations are on the z axis, but this time the grazing configurations are on the $z = 0$ plane, and the configurations with $\theta_i = \theta_e$, including the specular and retroreflective ones, are on the surface of the cone. Loci of fixed incidence and exitance coincide, and they are hemi-ellipsoids tangent to the surface of the cone.

5.7 BRDF Resampling

In order to make use of the measured BRDF samples, it is necessary to evaluate $f_r(\omega_i, \omega_e)$ at arbitrary points, not just at the sample points where the measurements took place. Because the samples are scattered arbitrarily over a 3D domain, this is a challenging reconstruction problem. In addition, the sampling density is anisotropic, because the spacing of the sample sheets is much sparser than the spacing of the samples within each sheet. To perform this reconstruction, we use a local polynomial regression method on the 3D domain defined by

ψ_3 in Section 5.6.3.

5.7.1 Local polynomial regression

The idea of local polynomial regression [19] is best introduced in one dimension. Consider a set of m irregularly spaced, noisy samples (x_i, y_i) of a function f, as shown in Figure 5.35. We want to define a continuous function $\tilde{f}$ that estimates what f might have been. One way to do this would be to define $\tilde{f}(x_0)$ to be the average of all y_i's corresponding to x_i's within a certain interval around x_0, yielding the function shown in part (a) of the figure. This technique could be called *local constant regression*, because we are performing regression, or data fitting, by using the value of the constant function that best fits the data in a local neighborhood of the reconstruction point. To obtain a continuous function, we can replace the simple average by a weighted average, with a weight that drops off with distance according to a *kernel* function h. This gives the result shown in part (b) of the figure. Using the same kernel, we can obtain a better fit with less tendency to reduce the height of peaks by fitting a line to nearby data instead of a constant function (part [c]), or, in general, a polynomial, as shown in part (d).

The basic computation underlying local polynomial regression is fitting a polynomial to the samples near x_0, using the weights given by $h(x_i - x_0)$. This can be done using a least squares system. For example, we can fit the quadratic function

$$p(x) = a_2 x^2 + a_1 x + a_0 \tag{5.4}$$

by finding the values of the parameters a_i that make $p(x_i - x_0)$ best approximate y_i for nearby x_i. Formally, we want to minimize

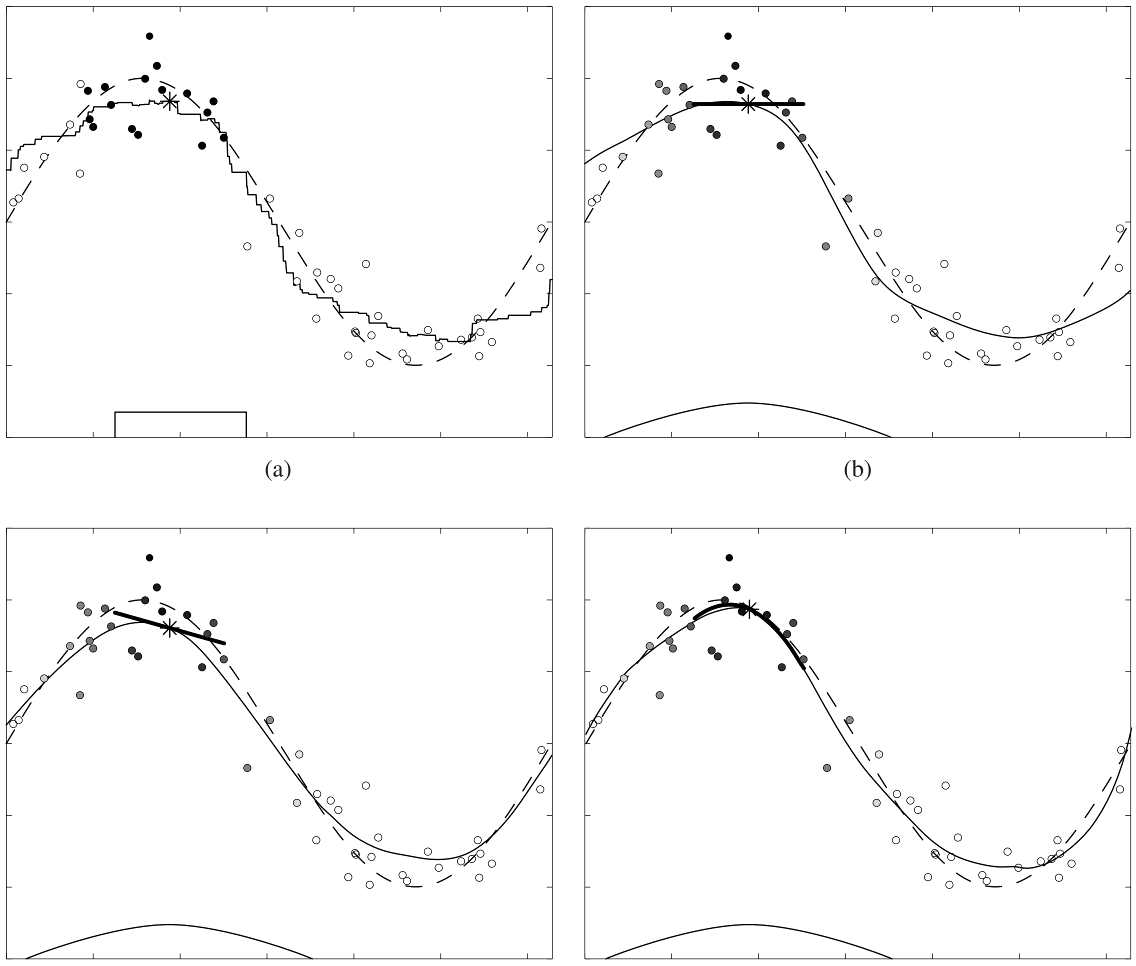
$$\sum_i \left[h(x_i - x_0)(p(x_i - x_0) - y_i)\right]^2 . \tag{5.5}$$

If we define the m by 3 matrix M to have entries $m_{ij} = h(x_i - x_0)(x_i - x_0)^{j-1}$, then this sum can be written as

$$\|Ma - y\|, \tag{5.6}$$

where $a = [a_0\ a_1\ a_2]^T$ and $y = [y_1\ \ldots\ y_m]^T$. This is a standard matrix problem that can be solved in time $O(n^2 m)$, where n is the number of coefficients in the polynomial, 3 for the 1D quadratic case. The entry a_0 is then $\tilde{f}(x_0)$.

The same idea can be applied in higher dimensional domains: for example, in the 3D case, local quadratic regression involves fitting the 10 coefficients of a quadratic in x, y, and z to the (at least 10, we hope) points that fall inside the support of the kernel. In three dimensions, the kernel can be chosen to be the same size on all three axes (a *spherical* or



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

Figure 5.35: Reconstructing a 1D function from irregular samples in four different ways. The kernel function h used in each case is plotted (on an unrelated vertical scale) at the bottom of the plot. The gray tone of each sample point indicates the value of h at that point, from white at zero to black at one.

isotropic kernel) or to have different extents on different axes (an *elliptical* or *anisotropic* kernel). If the desired elliptical kernel has the same shape and orientation everywhere, it can be implemented easily and efficiently by applying a linear transformation to the (x_i, y_i) and to x_0, then using a spherical kernel.

5.7.2 Reconstructing using ψ_3

If we transform our data points by ψ_3 before applying local polynomial regression, the sheets will lie approximately on parallel planes. This means we can bridge the gaps between the sheets without over-smoothing within the sheets by using an elliptical kernel that is elongated in the direction perpendicular to the sheets.

Very roughly, if there are 300 pixels across a full sheet, mapping to at most two units across the base of ψ_3's cone, and 30 sheets, mapping to $\sqrt{2}$ units on a 45° diagonal through the cone, we might expect to use a kernel with proportions of about 7:1:1. We chose the kernel size by following this proportion and making the kernel as small as possible without introducing points where there were too few samples within the kernel to make a reasonable estimate (such points show up as points where there are too few rows to solve the system, or as large spikes in $\tilde{f}$ if enough points fall within the kernel but they are nearly coplanar).

To produce the 3D scattering diagrams presented in Section 5.5, we used a simple adaptive subdivision scheme, beginning by evaluating f_r at the vertices of a tessellated hemisphere and repeatedly subdividing triangles with high curvature until all triangles were brought below a specified tolerance or a subdivision limit was reached.

This reconstruction technique, while it produces excellent results, is not well suited for direct use in a renderer, because of the large amount of memory required to store the samples and the high computational cost of fitting a polynomial for each BRDF evaluation.

5.8 Conclusion

This chapter has explained a simple technique that can measure the BRDFs of many materials and has described our implementation of the technique. We used only general-purpose imaging devices, but we achieved accuracy rivaling that of a specialized gonioreflectometer while measuring with greater speed and resolution than is normally possible. Our system measured datasets of roughly 1.5 million samples, and the resulting data were consistent and agreed closely with independent measurements.

The one major limitation of image-based BRDF measurement is that it can only measure materials that can be obtained in curved, homogeneous samples. Furthermore, for the measurement of full BRDF outside the incidence plane, we require samples with curvature along both axes. In practice, this limits us to homogeneous materials or coatings that can be sprayed or otherwise applied to a curved surface, or, in the case of incidence-plane measurements where we can use a cylindrical sample, materials that come in uniform, flexible sheets. While this limitation does exclude many surfaces, it includes many very important surfaces: paints and other coatings may be conveniently measured over the full BRDF domain, and the whole spectrum of paper and cloth may be measured in the incidence plane.

Though not an inverse problem in the traditional sense, computing a BRDF from images is very much a rendering task in reverse, requiring many of the algorithms and computations familiar from forward rendering. Ray tracing is used to find the surface configuration corresponding to each pixel; integration over the light source, camera aperture, and surface area defines the BRDF sampling; and interreflection is a factor for non-convex scenes.

Because it is impossible to control the pattern of samples that is measured, this method brings up interesting questions of how to understand complex sets in the BRDF's parameter space, and how to perform computations using them. We have described new ways of re-mapping the domain of an isotropic BRDF to aid in understanding the distribution of sample positions. We have also presented a novel mapping that puts the seemingly irregular sample pattern into a very convenient form that allows for much more efficient and high-fidelity resampling operations than would otherwise be possible.

5.8.1 Future work

There are a number of practical improvements and extensions that could be made to our method, both in the direction of increased measurement quality and in the direction of increased simplicity and decreased equipment cost.

We have demonstrated only measurements of isotropic materials. For the incidence plane, anisotropic materials may be measured using the system as already described, and the angle between the surface grain and the incidence plane may be varied by rotating the sample on the cylinder. To measure anisotropy on a spherical sample, the first requirement would be a spherical sample with a uniform anisotropic BRDF and known grain direction everywhere. One example of such a surface would be a metal sphere finished on a lathe with tool angle, speed, and feed rate relative to the surface maintained constant across the surface. If there is no requirement for internal checks on the data, only one eighth of the sphere's surface needs to be used for each measurement, so not all of the surface needs to be finished properly. The fourth degree of freedom in the measurement would then come from rotating the sample about an axis through its equator.

BRDFs of glossy surfaces have extremely high dynamic range, making them challenging to measure with any instrument. We applied a low-gloss finish to the normally glossy paints we measured, reducing the dynamic range to easily managed levels, but it should be possible to measure glossy surfaces by extending the dynamic range of the sensor using multiple exposure techniques. The finite source and detector solid angles would no longer be negligible in such a measurement, and they would have to be taken into account. The achievable dynamic range in images would be limited by flare in the primary camera optics and in the primary camera itself.

The measurements we have made with our system are already more than adequate for most purposes, so another interesting line of inquiry is how to reduce the cost or measurement time even further. The only significant costs in the system are the two digital cameras, and less expensive cameras could be substituted for either. The secondary camera, in particular, is a tool that is rapidly becoming a consumer electronics item, and a camera costing one tenth as much as the DCS 420 could easily be substituted today.

Reducing the cost of the primary camera must be done with care, since accurate and repeatable radiometric calibration is required. A camera without cooling would be much cheaper but would produce more noise, and a CCD array without the sophisticated anti-blooming measures used in the PXL camera's array would limit our ability to make up for that noise by extending the dynamic range with multiple exposures. However, if some additional noise and a reduced maximum dynamic range can be accepted, there is no reason the primary camera need cost more than one tenth what the PXL system cost.

The flexibility of this system could be increased by removing the need for precisely shaped samples, allowing pre-existing objects to be measured. Any convex surface that can be measured accurately can be used without changing the system, and a 3D range scanner is an obvious candidate for this measurement task. However, surfaces with truly uniform BRDFs that are *not* painted surfaces are rare, so a description of a pre-existing surface's BRDF will almost always require a spatially varying BRDF. A system that could measure spatially varying, nontrivial BRDFs on complex surfaces would build on the topics in this chapter and the previous one, and would be an invaluable tool in model acquisition.

Chapter 6

Conclusion

Work on three problems of *inverse rendering* has been described. Each uses the mathematical foundations of physically realistic rendering to solve problems in which the input is a photograph or photographs and the output is part of the scene description.

The first problem, *inverse lighting*, assumes knowledge of geometry, reflectance, and the recorded photograph and solves for the lighting in the scene. We have presented a formulation of the inverse lighting problem and a solution technique using a regularized linear least-squares system. This method has been demonstrated using both synthetic and measured input data, including photographs of human faces. We have also shown the results of a technique called re-lighting that makes use of the inverse lighting solutions to modify the lighting in the original photograph.

The second two inverse rendering problems solve for unknown reflectance, given known geometry and lighting. *Photographic texture measurement* concentrates on capturing the spatial variation in an object's reflectance. In our work, the geometric information came from scanned 3D models of real objects, and the image information came from multiple photographs with known lighting and camera characteristics. We have demonstrated software that uses this input to construct accurate, high-resolution textures suitable for physically realistic rendering. We have shown results both from synthetic data and from two complex natural objects with detailed surface textures.

The final project, *image-based BRDF measurement*, takes the opposite approach to reflectance measurement, concentrating on directional rather than spatial variation. In this case, we begin with simple, known geometry (spheres and cylinders) and spatially uniform reflectance, and we measure the full BRDF of the surface. The measurements are made by taking many photographs of an object with a light source in different positions, and we have demonstrated how these images can be made into high-quality BRDF measurements. We have presented the results of measuring several paints and one type of paper and have shown that the data have accuracy rivaling that of custom-built dedicated instruments. Because of our image-based approach and our novel light source positioning technique, our method requires only general-purpose equipment and is suitable for use outside of specialized optics laboratories. The cost of the apparatus is also much lower than the cost of a specialized gonioreflectometer. In addition, our method enjoys a major speed advantage: given an optimized setup, this technique could easily be used to build a system that measures orders of magnitude more points than present systems in orders of magnitude less time, when the wavelength spectrum of the BRDF does not need to be measured in detail.

The investigation of these three problems in inverse rendering has raised many new questions even as it has answered others. In Chapter 3, we demonstrated the ability to infer plausible lighting from photographs. This shows that inverse lighting is possible for some models; clearly it is impossible for others, such as a planar surface. What can be said about which models and which BRDFs lead to reliable, stable inverse lighting solutions? In the presence of shadows and interreflections, this is a complex, but fundamental, issue.

In Chapter 4, we showed how high resolution texture maps can be assembled for objects with very complex geometry. We have also investigated ways to construct the same types of texture maps for surfaces with non-trivial BRDFs. How does interreflection affect the results? With interreflection, how does the complexity of the problem increase as we allow a BRDF to approach mirror-like behavior? Here we see the difficult-to-constrain long-distance interactions that make rendering such a difficult problem.

In Chapter 5, we found that surprising accuracy can be obtained from a simple camera-based measurement system. Clearly, this can become an important measurement technique for the materials to which it is applicable. Can it be made to apply outside the constrained laboratory setting in which it has produced these results? Could we measure BRDFs under uncontrolled illumination?

Chapters 4 and 5 are, as we have pointed out, addressing two aspects of the same problem. To produce the models best suited for physically realistic rendering, we need a tool that can capture all characteristics of an object that have an important effect on rendered images. This goal lies somewhere between the achievements of our two systems. The resolution and generality of our BRDF measurement system neither can be nor need be obtained independently at every point, but for realism and especially for physical accuracy we must robustly account for the full BRDF. To combine these two systems' strengths is an important challenge in the effort to characterize objects for rendering.

Appendix A

Camera Calibration

The calibration of cameras for measurement breaks down into two parts: *geometric calibration*, which establishes the relationship between image points and rays in 3-space, and *radiometric calibration*, which establishes the relationship between pixel values and radiance in the scene.

A.1 Geometric Calibration

The two parts of geometric calibration are finding the camera's pose, or its location and orientation, called the *external parameters*, and finding all the characteristics of the camera itself, the *internal parameters.*

Cameras are commonly described by the pinhole perspective model, with deviations from that ideal behavior modeled as geometric lens distortions [22, 21, 20, 60]. According to that model (Figure A.1), a point $\mathbf{x}$ in 3D space is imaged on the *image plane* P of the camera at the point $\mathbf{y}$ that is collinear with $\mathbf{x}$ and the camera's *center of projection*, $\mathbf{c}$. The image that we read out of the camera is measured from a rectangle R on the image plane.

We describe cameras starting with a coordinate frame based at $\mathbf{c}$, with its w axis perpendicular to, and pointing towards, P and its u and v axes parallel to the edges of R. This coordinate frame, with its six degrees of freedom, comprises the external parameters of the camera. For a sensor of given dimensions, the internal parameters are the w coordinate of the image plane, called the *principal distance*, and the position of the image rectangle. The position of the image rectangle is specified by giving the location (p_u, p_v) of the *principal point*, the foot of the perpendicular from $\mathbf{c}$ to P, in the coordinates of the image. Thus a pinhole perspective camera has in total nine degrees of freedom, for a particular sensor size.[1]

[1]Note that the pinhole model ignores the orientation of the lens, so effects due to the lens, such as off-axis irradiance falloff, need not be centered at the principal point if the lens axis is not

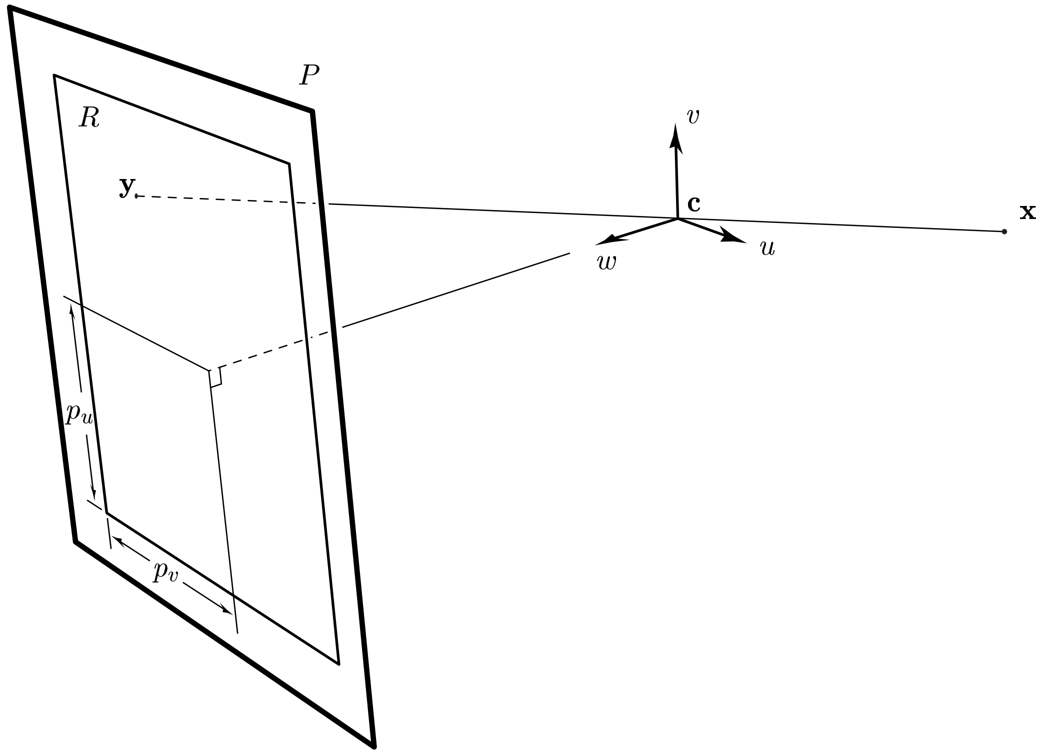


Figure A.1: The pinhole camera model.

With the high-quality lenses and narrow fields of view that we used, we found it unnecessary to include geometric distortions in the camera model, so we used the pinhole perspective model directly.

To measure camera parameters, we used correspondences between 3D points at known locations and their projections in the image. In the earlier work described in Chapter 4, we used a box with grid targets on its surfaces to provide the target points, which were located by manually clicking on the intersections of grid lines in enlarged images. Later on, we instead used the automatically recognizable targets described in Appendix C. The 3D locations of the target points were found in one of two ways: by locating the target points in the luminance channel of range images from the Cyberware scanner (which give the 3D coordinates of the corresponding points directly in the scanner's coordinate system) or by using the technique of bundle adjustment described in Appendix B to locate a set of targets from a series of photographs.

Given a set of 3D points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and their projections $\mathbf{y}_1, \ldots, \mathbf{y}_n$, we can find the parameters of the camera by solving a fitting problem. We encapsulate the camera's parameters in a vector with nine entries to account for the nine degrees of freedom in the

perpendicular to the image plane.

camera model.[2] Let $f(\mathbf{a};\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$ be the camera model for the parameter vector $\mathbf{a}$; that is, $f(\mathbf{a};\mathbf{x})$ is the location that the camera model predicts for the image of the point $\mathbf{x}$ using the parameters contained in $\mathbf{a}$. We can estimate $\mathbf{a}$ by solving the following nonlinear least squares problem:

$$\min_{\mathbf{a}} \sum_i (f(\mathbf{a};\mathbf{x}_i) - \mathbf{y}_i)^2. \tag{A.1}$$

We have written code in the MATLAB numerical computation environment to solve this system with some or all of the internal parameters held fixed. As with many nonlinear equations, finding a good starting point for the iterative solution is important for quick convergence to the correct answer. We use a weak perspective approximation to the camera model for this purpose [20]. Because of the comparatively small amount of data we get from a single view, we normally do not attempt to find all nine parameters with this equation; rather, we solve for the pose alone (6 parameters) or for the pose and principal distance (7 parameters), using previously measured values for the internal parameters that we hold fixed.

The precision of the result of this pose estimation process depends on the set of points being used, what internal parameters are variable, and the precision of the input measurements. A strong point set is one that has points at a range of distances from the camera; this allows two types of nearly indistinguishable motions to be measured more accurately: first, it helps distinguish camera translation perpendicular to the view direction from camera rotation, and second, it helps distinguish translation along the view direction from change in focal length.

To find the full set of internal parameters, we used the self-calibrating bundle adjustment computation described in Appendix B. By this means we measured the internal parameters once for each camera-lens combination we used. We assumed that the principal point so obtained was valid for any focus setting on the lens, but the principal distance clearly changed with focus setting and therefore had to be re-measured each time that setting was changed. Since there was no way to precisely repeat a focus setting, this meant measuring the focal length before every experiment. In situations where camera pose was being estimated from a geometrically strong set of point correspondences, we included the focal length in the pose estimation, finding it simultaneously with the camera pose, but in situations where the point set was weaker, we used a small bundle adjustment system with fixed principal point to estimate the focal length, which was then used as a fixed value in the pose estimation.

[2]The pose is represented by the cartesian coordinates of $\mathbf{c}$ and a Cayley transformation [61] representation of the orientation of the frame.

A.2 Radiometric Calibration

Whenever we interpret pixel values in images as radiance measurements, we must know the relationship between the radiance arriving at the camera's lens and the digital code values reported by the camera software. The function that maps radiance to digital code value is known as the *opto-electronic conversion function*, or OECF. It can vary from pixel to pixel, both because of manufacturing variations in the CCD elements themselves (known as fixed-pattern noise) and due to variations with angle in the radiance throughput of the camera (known as lens falloff) [33, 32]. We treated radiometric calibration differently for each of the two cameras described in Appendix D.

The PXL camera system is designed to have a linear OECF, and that property has been separately verified for the particular camera system we used [13]. The task of radiometric calibration for this camera is to characterize the variation in the slope and offset of this linear function across the image, an operation known as *flat field calibration*. We compensated for pixel-wise variation in offset by subtracting a dark image, taken without opening the camera's shutter, from each image we measured. We measured the slope variation by exposing the camera to the uniform radiance field produced by an integrating sphere light source (Labsphere CSTM-USS-1200),[3] using several focus and aperture settings. For each image so obtained, we fit the radial quadratic model

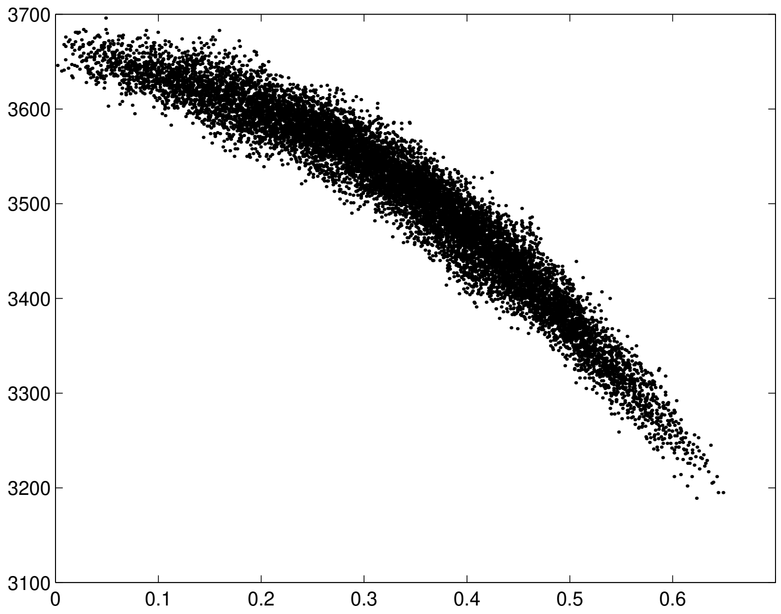
$$a - b[(x - x_0)^2 + (y - y_0)^2] \tag{A.2}$$

to the pixel values, solving for a, b, x_0, and y_0. The resulting values for a and b varied systematically with focus but not with aperture over the range we used ($f/8$–$f/22$). The resulting values for x_0 and y_0 also varied with focus, though barely measurably; we used a single (x_0, y_0) pair for all focus positions with no appreciable effect on the results. Dividing by the fitted quadratic model corrected for all noticeable systematic variation in response. For example, Figure A.2, part (a), shows the uncorrected pixel values for one image plotted against distance from (x_0, y_0). Note how the values decrease away from the center of the image. (The few outliers are caused by dust in the camera.) The corrected values are plotted on an expanded scale in part (b).[4] The remaining variation, due to random and fixed-pattern noise (including that due to dust in the optical system), is less than 1%, which we considered acceptable for our purposes.

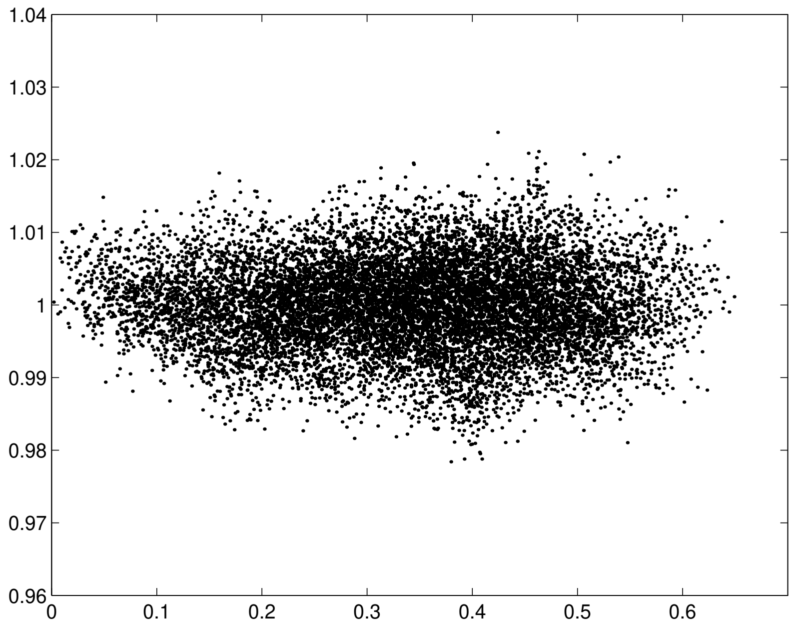
For the DCS camera, we assumed lens falloff to be negligible, because of the longer focal lengths and smaller sensor dimensions involved compared to the PXL camera. We also ignored fixed-pattern noise. We characterized the OECF by photographing a calibrated

[3] The sphere was illuminated by the SB-16 flash, rather than by its built-in sources.

[4] The plots actually show only a fraction of the pixels, because including all the pixels makes the figure hard to read.



(a)



(b)

Figure A.2: Correcting for lens falloff in flat-field calibration. Pixel values versus distance from image center (a) before correction and (b) after correction.

reflectance step target (Vertex Video Systems Accu-Chart EIA Standard Reflectance Chart) with patches of 9 different reflectances and fitting the function $y = k(x + c)^{\gamma}$ to the known reflectances and measured pixel values.[5]

To establish the absolute scale of the measurements from both cameras, we measured the reflection from a sample of known diffuse reflectance (a Spectralon sample manufactured by Labsphere, $R = 0.99 \pm 0.002$ over the visible range) with normal incidence and known light source position, using the same filters and lens settings that were used during the actual measurements. Before we stabilized the light output from the flash (Appendix D), we included the sample in every measurement image, but with the light source stable we only needed to calibrate once.

[5]The DCS420's sensor has masked columns at the sides of the image for measuring dark current; we subtracted the mean value of these dark pixels from each image before using it.

Appendix B

Bundle Adjustment

To calibrate cameras and to measure the positions of targets used for later pose estimation, we used a technique known in the photogrammetry literature as *self-calibrating bundle adjustment* [26, 22, 12]. The idea of bundle adjustment is as follows: suppose we have taken n photographs of m fixed three-dimensional points $\mathbf{x}_j$. If we record the position $\mathbf{y}_{ij} = (u_{ij}, v_{ij})$ of the image of point j in camera i for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ (Figure B.1), we get a total of $2mn$ measurements that depend on both the poses of the n cameras and the locations of the m points. Together, the camera poses and point locations have $6n + 3m$ degrees of freedom. For sufficiently large m and n, $2mn > 6n + 3m$, so we can hope to solve for all the camera poses *and* all the point locations starting with nothing but the camera model and the measured image coordinates.[1]

Following the notation of Appendix A, we can write the equations to be solved as:

$$f(\mathbf{a}_i; \mathbf{x}_j) = \mathbf{y}_{ij} \qquad i = 1, \ldots, n; k = 1, \ldots, m. \tag{B.1}$$

This system is overdetermined for large enough m and n, so we solve it as a nonlinear least squares problem with $6n + 3m$ variables and $2mn$ observations. We use the Levenberg-Marquardt algorithm, as implemented in the MATLAB numerical computation environment, with the perspective camera model described in Appendix A.

Finding a good starting point is critical for quick and correct convergence. We use the elegant linear approximation proposed for the shape-from-motion problem, which is essentially identical to bundle adjustment, by Tomasi and Kanade [58]. Their algorithm takes the same form of input as ours, and because the formulation is linear its unique solution can be found directly, without using iterative minimization algorithms.

[1] Our practical ability to do this will depend on the particular sets of points and camera poses—for example, if all the points are collinear or all the cameras are at the same location, the system will degenerate and will fail to have a unique solution.

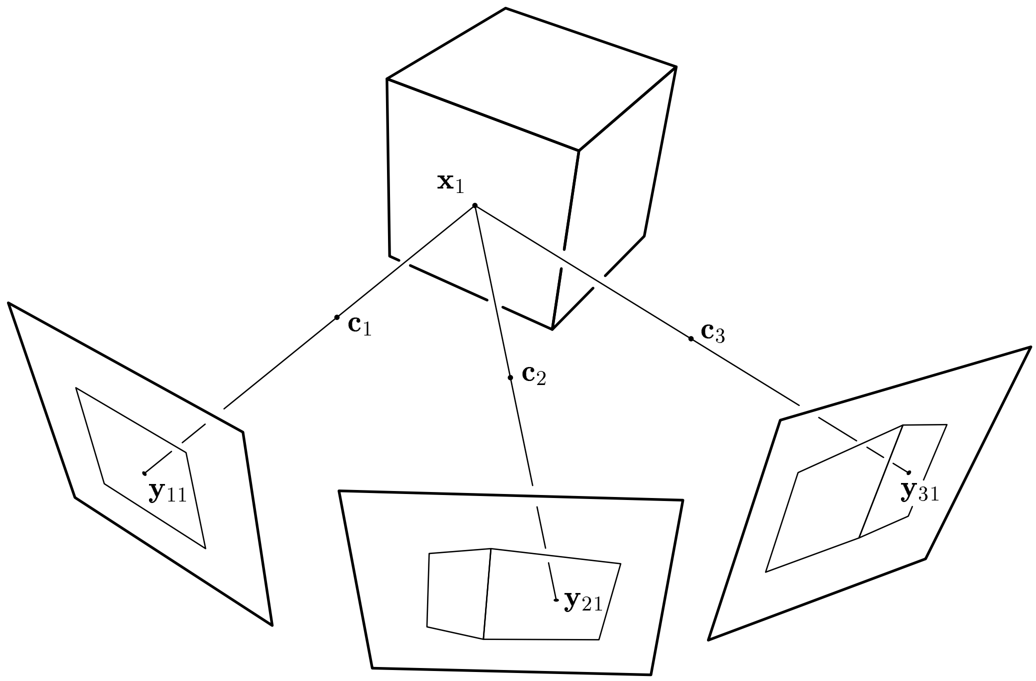


Figure B.1: Notation for bundle adjustment.

The solution of the system is slightly complicated by cases in which not all points are visible to all cameras. Omitting some equations from the system in Equation B.1 is simple, but extending the linear approximation to accommodate incomplete data is less so. Tomasi and Kanade do describe a technique to handle missing points, but it is fairly involved. Instead, we simply use a subset of the points and cameras for which visibility is complete to compute the initial values for many of the parameters, then use pose estimation (Appendix A) and triangulation to fill in starting values for the rest of the cameras and points.

Since the system is overdetermined, we can include more variables. If the internal parameters of the cameras are unknown, we can include them as variables and solve for n camera poses, m point locations, and p internal parameters. This technique is known as *self-calibrating bundle adjustment*, since the camera calibration is handled as part of the computation and need not be done separately. In fact, self-calibrating bundle adjustment can be used as a camera calibration technique, in which case the camera poses and point positions are computed as a side effect of finding the internal camera parameters.

The precision of the results computed by bundle adjustment depends on the precision of the input points and the sensitivity of the output to perturbations in the input. This sensitivity depends on the configuration of points and cameras and on what internal parameters are included in a self-calibrating system.

Appendix C

Calibration Targets

The pose estimation and bundle adjustment algorithms described in Appendices A and B require correspondences between known 3D points and their projections in images. To provide these correspondences reliably and precisely, we developed special targets, printed on a laser printer, that were automatically located and identified in images.

The basic design of a target, shown in Figure C.1, is a solid circle surrounded by a ring of curved bars. The circle is used to establish the position of the target and to estimate its distance and orientation. The surrounding bars encode an 8-bit identifying number (the target's *ID*) that allows the target to be distinguished from other targets. The code also serves as a check to prevent other objects from being recognized as targets, since a non-target is unlikely to be surrounded by a valid ID code.

Like many bar codes, our target IDs encode binary numbers in the widths of the black and white bars (the white bars are the spaces between the black bars), with a wide bar denoting 1 and a narrow bar denoting 0 (Figure C.1). All the widths are integer multiples of a basic angular *unit*: a narrow bar is one unit wide, a wide bar two units, and the long white bar that marks the start of the code is at least three units wide. Because readability under uncertain focus and lighting was important to us, we used the largest unit that could encode enough distinct IDs for our purposes, but in applications needing greater numbers of targets a smaller unit could be used to encode more bits.

To avoid misrecognizing targets that are too small, too oblique, or partially obscured, and to avoid recognizing other objects as targets, we put fairly stringent requirements on what will be reported as a target. Every estimated bar width must be near an integer multiple of the unit, and the total of all the bar widths must account for the entire circle. There is also a parity bit to detect any errors that creep through. Because the most common error in reading these codes is a two-bit error caused by a boundary shifting far enough from its proper position to make a short-long sequence appear as a long-short sequence, the parity is computed on only the even-numbered bits, rather than on all the bits.

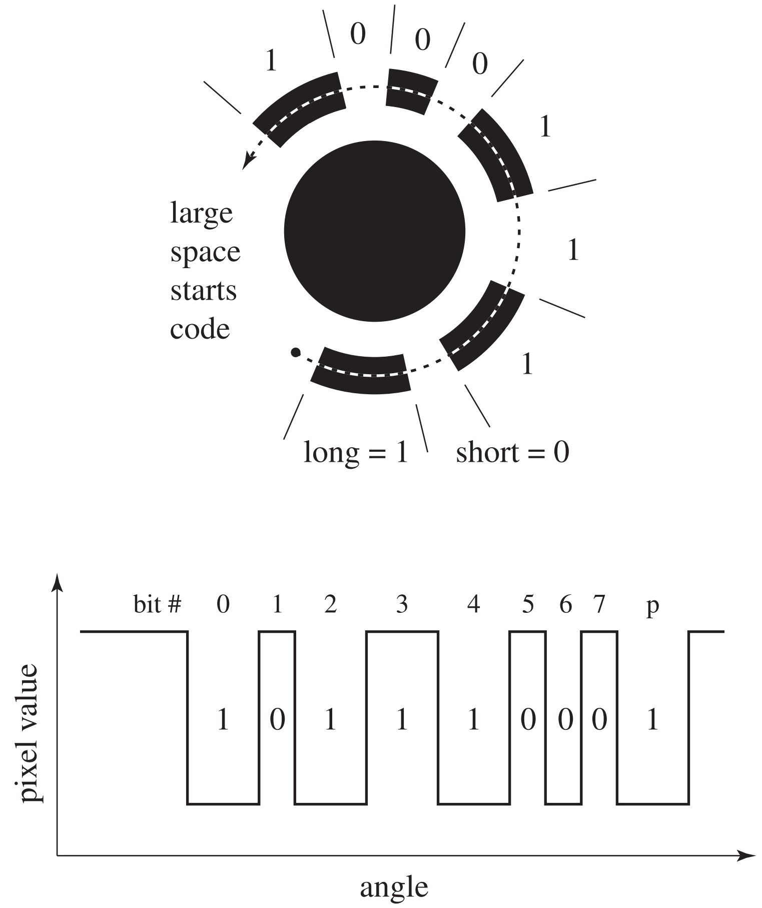


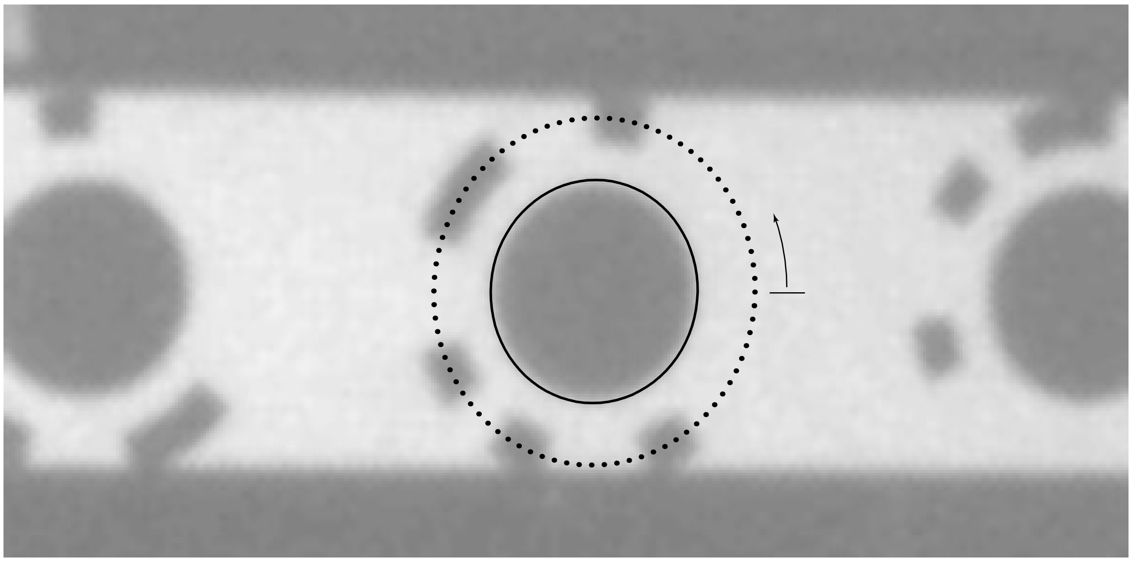
Figure C.1: The meaning of the ID code printed around a target.

The algorithm to recognize all the targets in an image is as follows. The values t, a_l, a_h, d, and α are parameters to the algorithm.

Find all contiguous regions of pixels with values below t that have
    areas between a_l and a_h.
FOREACH such region R:
  Measure the region's shape:
  Compute the moments of R up to second order, including the major
    and minor axis lengths a and b and the center of mass (x_c, y_c).
  Compute the ellipse E that fits R's moments.
  Find the bar code ring:
  Let E' be the ellipse concentric with E and with major and minor
    radii enlarged by α.
  Sample the image around E', four samples per unit.
  Verify and decode the target ID:
  Verify that the dynamic range of the collected samples is at least d.
  Divide those samples into white and black spans.
  Classify the spans by length as follows:
    3–5 samples $\Longrightarrow$ short bar
    7–9 samples $\Longrightarrow$ long bar
    11 or more samples $\Longrightarrow$ start marker
    any other length $\Longrightarrow$ ID is unreadable
  Verify that the start marker is a white bar.
  Decode the ID from the sequence of long and short bars.
  Check parity using the last bit.
  Report (x_c, y_c), A, and the ID.

When the samples are separated into spans, we use a dual-threshold hysteresis technique. High and low thresholds are computed from the 10^{th} and 90^{th} percentile sample values; the low threshold is $.6v_{10} + .4v_{90}$ and the high threshold is $.4v_{10} + .6v_{90}$. As we step through the samples, a transition from low to high is registered when the high threshold is crossed, and a transition from high to low is registered when the low threshold is crossed. This is meant to prevent noise from causing extra transitions. Figure C.2 shows the ellipse E and the sample points for a noisy target image from a BRDF measurement experiment. It also shows the resulting sample values and how they were classified as white (open circles) and black (filled circles). The ID is decoded from the sequence of bars, starting with the bar after the start marker, by building a binary integer with a zero for each short bar and a one for each long bar.

To recognize targets in a variety of lighting conditions, we run this algorithm for several values of t and α. The threshold t depends on lighting because lighting affects the overall intensity of the image, but the size ratio α also depends on lighting because, especially for poorly focused targets, the relationship of t to the image intensity affects the estimated size of the ellipse (though not its center, which is the primary measurement reported by this



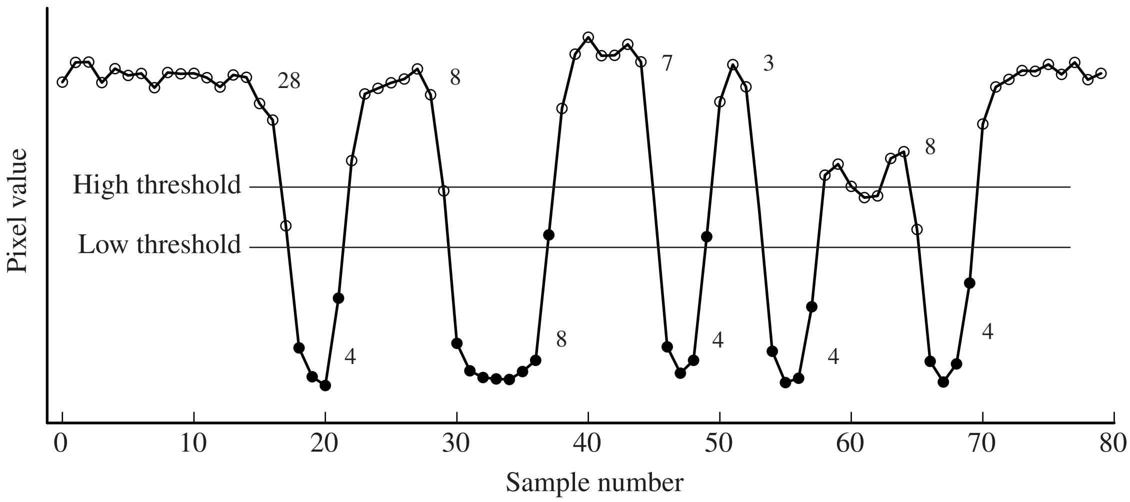


Figure C.2: A noisy target image with the best-fit ellipse and the sample points marked. Below, a graph of the resulting sample values, showing how they are classified into black and white. The number of samples counted toward the width of each bar is noted next to the corresponding part of the graph.

algorithm). Therefore trying several values of α helps find the bar code when the target is difficult to read. When a target is recognized more than once, the median of the reported coordinates is used as the final measurement.

This algorithm has been through several revisions. Earlier versions rarely misrecognized targets (one out of 300 was typically in error), and all targets that have ever been misread would have been correctly rejected by the most recent version. Most errors were caused by dark shadows falling across targets and lengthening the black bars of the ID code. Versions that used a standard parity computation were susceptible to bit reversals (10 to 01 and vice versa) under these conditions, but the later even-bit parity scheme detects such errors. Single bit errors have never been observed.

The accuracy of the target locations depends on their size and on the uniformity of illumination. An illumination gradient across the target, particularly for a poorly focused image, can skew the position toward the darker side. However, since our experiments have all used flash illumination from the camera position, errors caused by illumination variations have not been a problem. By analyzing several DCS420 images from the same camera position, we estimated the repeatability of locating targets to be approximately 0.5 μm, or 1/20 of a CCD pixel, under realistic operating conditions representative of the situations encountered in our experiments.

Appendix D

Cameras

The images that served as input to the inverse rendering systems described in this dissertation came from two digital still cameras.

The Photometrics PXL 1300L is a scientific digital camera designed to make precise image measurements, with low noise even over long exposure times. Its image sensor is a 1280 by 1024 pixel Kodak KAF-1300i CCD array. The sensor is cooled thermoelectrically, and a forced liquid coolant dissipates the heat; the array was operated at a temperature of $-25°$ C. This cooling reduced the dark current to two electrons per pixel per second (as measured by the manufacturer), which is immeasurable for the short exposure times we used. The pixels are square, measuring 16 μm on a side, leading to a total array size of 20.48 by 16.38 mm. The same lens was used for all of the experiments, a 35 mm format Nikkor lens with a nominal focal length of 28 mm. All aspects of the camera's operation were controlled through the IPLab software system running on an Apple Macintosh computer.

Since the PXL camera is a monochrome device, when we needed to measure wavelength dependence we used color separation filters. We used the red, green, and blue filters from a filter wheel that was originally part of an Eikonix camera. They appear to be multilayer coated interference filters, but their manufacturer is not known. We measured their spectral transmittance using an Optronic OL-750 spectrometer; the resulting curves are in Figure D.1. Because they have significant unwanted transmittance in the infrared, where the camera is very sensitive, we used an additional filter, Oriel #57400, to block the near infrared portion of the spectrum. The spectral transmittance of that filter is also shown in Figure D.1.

When we needed to reduce the light sensitivity of the camera, we placed neutral density (ND) filters in front of it. The ND filters we used are from the set Melles Griot #03 FSQ 015, and they are made of metal-plated glass.

The other camera we used was a color Kodak DCS 420. This camera is meant for the professional photography market, and is widely used by photojournalists. As such, it

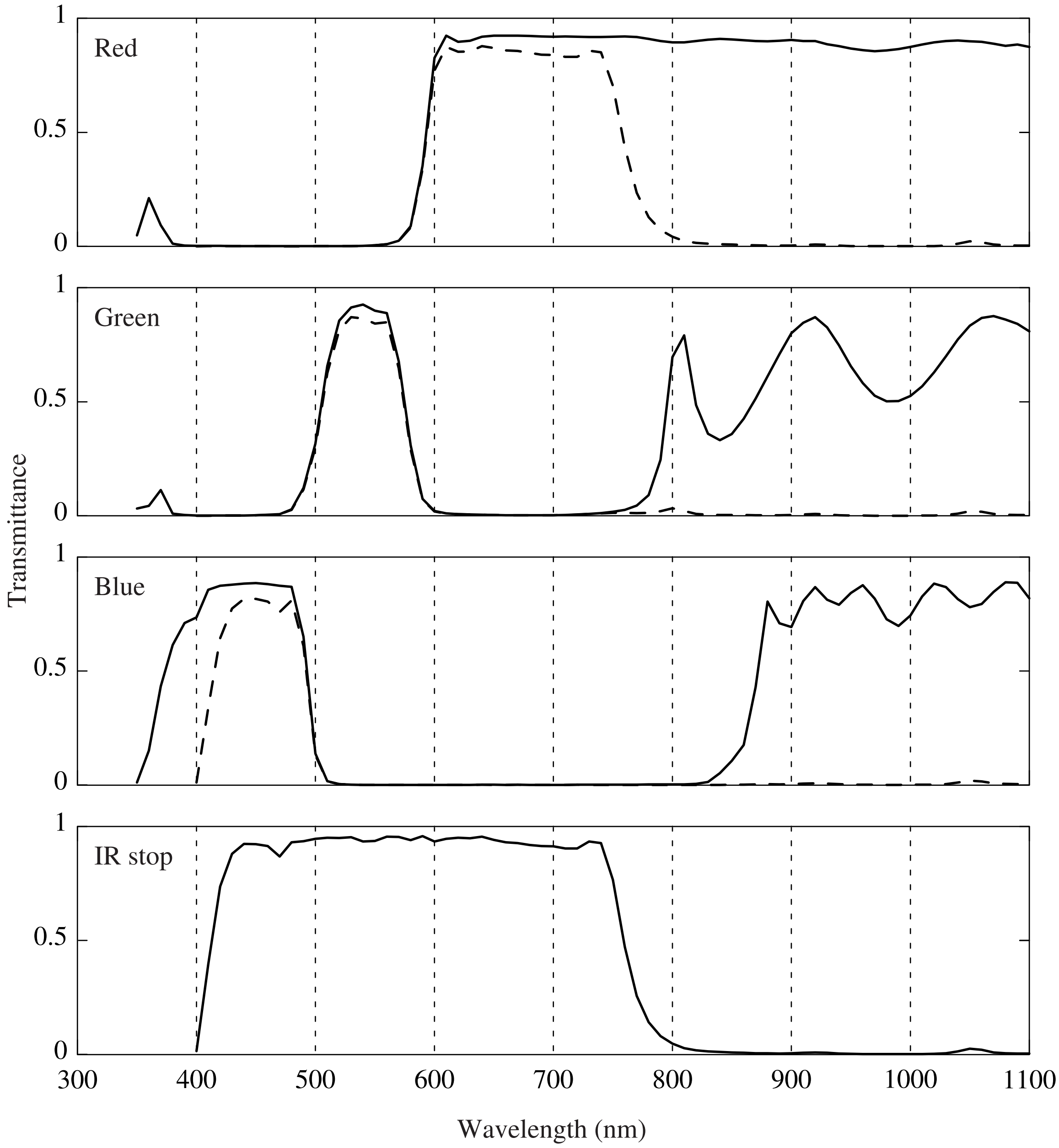


Figure D.1: Spectral transmittance of filters used with PXL camera. Plots are shown for the three color separation filters and the filter used to block the near infrared. The product of the IR filter's spectrum with each of the color separation filters' spectra is shown as a dashed line.

concentrates less on low noise and more on convenience and portability. Kodak's digital camera back fits onto a Nikon N90 camera body, replacing the film with a CCD array; the N90's lens and shutter, as well as its autofocus and autoexposure systems, operate just as they would when exposing film. The sensor is a Kodak KAF-1600c, which has 1536 by 1024 9 μm pixels (image area 13.8 by 9.2 mm) and an RGB color filter array. When we needed color images, we used Kodak's supplied software to read the images from the camera and reconstruct a full-color image from the color filter mosaic, but when we needed only greyscale we began with the raw images and linearly interpolated the green pixels to a full image. We used two lenses with this camera, both 35 mm format Nikon lenses: a 50 mm standard lens and a 55 mm macro lens.

The illumination for all the photographs from both cameras was provided by a Nikon SB-16 flash, which incorporates a xenon flash tube, together with electronics for various auto-exposure functions. We used the flash exclusively in Manual mode, in which it produces a full-power flash every time it is triggered.

When consistency from flash to flash was required, we powered the SB-16 from a regulated power supply rather than from the supplied battery pack. Our experiments indicate that with regulated voltage and a charge time of at least 20 seconds, the mean-square variation in output is less than 1.5%. When we used the flash with the DCS 420 camera, it was triggered by the camera electronics through the contacts in the camera's flash-mounting shoe.

It is also important for the light output to be angularly uniform. By using the flash to illuminate a uniform white board and measuring the reflection with the already calibrated PXL camera, we concluded that the light output is uniform to within 5% over a circle 10° in diameter. This is much larger than the range of angles we used in our experiments.

We measured the spectral energy distribution of the flash's output using an Oriel Multispec 77400 spectroradiometer. The distribution is shown in Figure D.2.

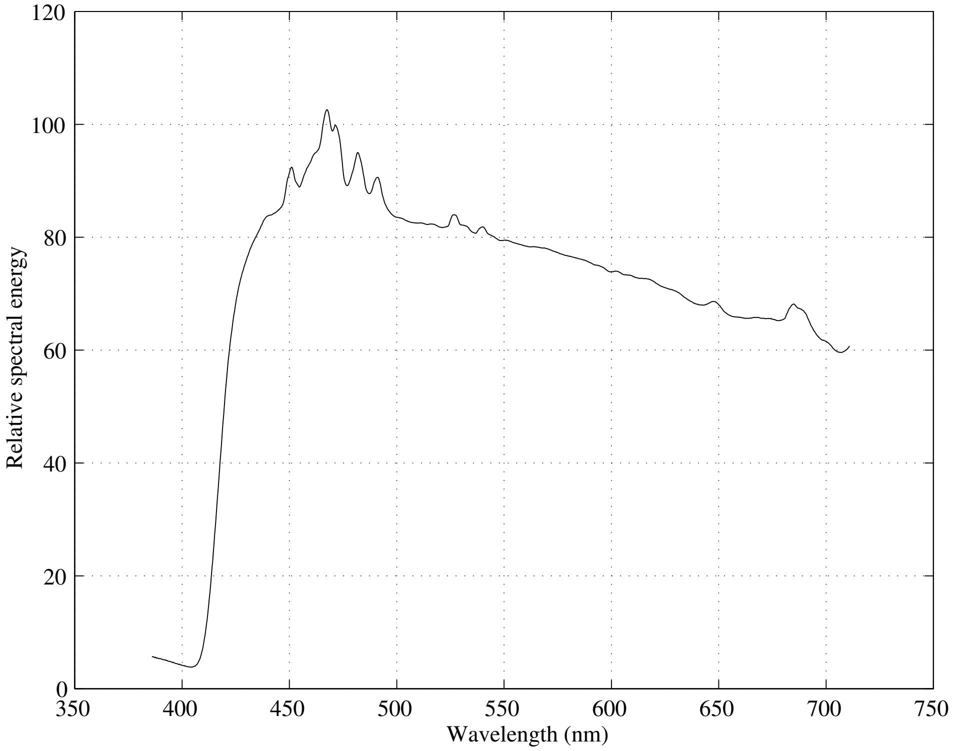


Figure D.2: The spectral energy distribution of the SB-16 flash.

Appendix E

The Cyberware Scanner

The Cyberware 3030 is a structured-light range scanner. It senses distances to a surface by projecting a stripe of light and observing the reflection of that stripe with a video camera.

A laser produces a parallel beam, which is fanned out by a cylindrical lens into a vertical sheet of light. This sheet defines the *scanning plane*. Any diffuse surface in front of the scanner reflects light from the curve where it intersects the scanning plane. A monochrome CCD camera positioned at an angle records an image of that curve, and since the curve is known to lie in the scanning plane the 3D position of the portion of the curve corresponding to each row of the camera image can be computed. This process leads to several hundred 3D points that describe the shape of the surface where it intersects the scanning plane.

To scan a 2D surface, the scanner moves relative to the surface so as to sweep the scanning plane through a volume of space containing the object. The scanner acquires 30 curves per second, so that all of the surface visible to the scanner can be digitized over the course of a few seconds.

The scanner is moved relative to the object by one of two motion platforms. The PS platform, for scanning human subjects, consists of a platform, on which a chair is placed where the subject sits, and an arm that holds the scanner at head height and rotates a full $360°$ around the platform. The resulting dataset consists of a series of constant-θ curves in an (r, y, θ) cylindrical coordinate system.

The second platform, the MS platform, is meant for scanning rigid objects. On this platform the scanner stays stationary while the object either rotates or translates in front of it. The object sits on a 600 mm circular turntable, which is mounted on a 1.5 m horizontal translation stage. Only one of the two possible motions is used for any given scan, leading to data in cylindrical coordinates for a rotational scan or in Cartesian coordinates for a translational scan.

The data from the scanner consists of a 2D array of points, together with luminance values that give the strength of the reflected signal used to locate each point. This luminance image can be used to locate targets on objects being scanned for calibration. The scanner also includes a separate RGB camera to provide measurements of surface color, but we opted to use color data from a separate camera when we needed it.

Appendix F

BRDF Measurement Procedure

This appendix gives the details of the procedure used to obtain the data reported in Chapter 5.

The samples were prepared for measurement by painting them with consumer spray paints. Achieving a uniform finish while painting was critical to having a well-defined BRDF to measure. The cylinder was painted while standing on a flat surface; the spheres were painted while supported in mid-air on a stand. Two to three coats of each paint were applied, with the spherical samples rotated between and during coats. At the same time and under the same conditions, flat aluminum plates were painted to be measured independently. In the case of the cylinder, care was taken to record the orientation of the flat sample so that it could be measured in the same plane as the cylinder to prevent any gravity-induced anisotropy from affecting the comparison.

After the samples dried, they were brought into the lab and measured. The procedure used for a typical measurement, including the calibration steps required for a particular sample setup, is as follows.

1. *Center the test sample on the turntable.* Place the sample near the center of the turntable.[1] Set up a laser to graze the surface and rotate the table, repeatedly adjusting the sample and the laser until a full revolution can be made with the laser grazing the surface.

2. *Set up the primary camera.* Set the desired lens aperture (normally $f/11$). Approximately level the camera rail using the tripod's controls. Focus on the silhouette edge of the test sample.

3. *Set up the secondary camera.* Set the desired lens aperture to avoid saturation at

[1] The turntable axis was known relative to the calibration targets because some of the targets were attached to the edge of the circular turntable.

the closest approach to the calibration targets (normally $f/32$). Set the focus at an intermediate value (normally 2 m). Set all camera modes to manual, and set the camera for minimum sensitivity (ISO 100).

4. *Set up the light source.* Attach the flash to the secondary camera. Connect it to the power supply, and set the supply limits at 6.00 V, 5 A. Set the flash to manual mode. Set the height of the secondary camera's tripod to put the light source at approximately the same height as the primary camera's lens.

5. *Calibrate the primary camera's pose.*

   (a) Put a temporary set of calibration targets (laser printer output glued to a metal plate) in front of the test sample.

   (b) With the translation platform at 0 mm (all movements of the platform are made relative to a fixed "home" position, designated 0 mm), take three exposures with the secondary camera from well-separated viewpoints. Be sure that all the temporary targets and several permanent targets are visible in each image.

   (c) Take three exposures with the primary camera, with the translation stage at -100 mm, 0 mm, and 100 mm.

   (d) Return the motion stage to 0 mm and remove the temporary targets.

6. *Make the measurements.* For each of a number of light source positions (normally 32), do the following:

   (a) Move the secondary camera tripod to the next position.

   (b) Pan the secondary camera to face the sample and tilt it so that it can see the calibration targets below the sample.

   (c) Start a 0.5 second exposure on the primary camera, and manually trigger the secondary camera between the shutter clicks.

   (d) If color separation filters are being used, repeat the exposure for the other two colors (trigger the flash directly for these exposures rather than triggering the secondary camera, so as to end up with only one calibration image per camera position).

Throughout the measurement, care was taken to ensure that light from the flash could not directly enter the box around the primary camera's optics, because such direct illumination can introduce stray light, reflections, and lens flare into the measurement images. These problems were prevented by setting up a black baffle to shadow the camera and by moving it from time to time as required by the changing light source position.

Bibliography

[1] Adobe Systems, Inc. *Adobe Photoshop 5.0 User Guide.* San Jose, CA, 1998.

[2] M. Agrawala, A. C. Beers, and M. Levoy. 3D painting on scanned surfaces. In *1995 Symposium on Interactive 3D Graphics*, pages 145–150. ACM SIGGRAPH, April 1995.

[3] Suraiya P. Ahmad and Donald W. Deering. A simple analytical function for bidirectional reflectance. *Journal of Geophysical Research*, 97(D17):18,867–18,886, 1992.

[4] J. M. Airey, J. H. Rohlf, and F. P. Brooks, Jr. Towards image realism with interactive update rates in complex virtual building environments. In *1990 Symposium on Interactive 3D Graphics*, pages 41–50. ACM SIGGRAPH, March 1990.

[5] James Arvo and David Kirk. Particle transport and image synthesis. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, pages 63–66, August 1990.

[6] James Richard Arvo. *Analytic Methods for Simulated Light Transport.* PhD thesis, Yale University, 1995.

[7] Réjean Baribeau, Marc Rioux, and Guy Godin. Color reflectance modeling using a polychromatic laser range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):263–269, 1992.

[8] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 35–42, July 1992.

[9] Chakib Bennis, Jean-Marc Vézien, and Gérard Iglésias. Piecewise surface flattening for non-distorted texture mapping. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 237–246, July 1991.

[10] Eric A. Bier and Kenneth R. Sloan, Jr. Two-part texture mappings. *IEEE Computer Graphics and Applications*, 6(9):40–53, September 1986.

[11] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–546, 1976.

[12] J. H. Chandler and C. J. Padfield. Automated digital photogrammetry on a shoestring. *Photogrammetric Record*, 15(88):545–559, 1996.

[13] Steve Shiang-Feng Chen, Jerry Wei-Chieh Li, Kenneth E. Torrance, and S. N. Pattanaik. Preliminary calibration of the Photometrics PXL1300L CCD camera. Technical Report PCG-96-1, Cornell University Program of Computer Graphics, 1996.

[14] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, Massachusetts, 1990.

[15] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 303–312, August 1996.

[16] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 11–20, August 1996.

[17] Julie Dorsey, James Arvo, and Donald Greenberg. Interactive design of complex time dependent lighting. *IEEE Computer Graphics and Applications*, 15(2):26–36, March 1995.

[18] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 173–182, August 1995.

[19] J. Fan and I. Gijbels. *Local Polynomial Modeling and Its Applications.* Chapman & Hall, London, 1996.

[20] Olivier Faugeras. *Three-dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, Massachusetts, 1993.

[21] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, Massachusetts, second edition, 1990.

[22] C. S. Fraser, M. R. Shortis, and G. Ganci. Multi-sensor system self-calibration. In *Videometrics IV*, pages 2–18. SPIE, October 1995. Invited paper.

[23] A. Gagalowicz and Song De Ma. Model driven synthesis of natural textures for 3-D scenes. *Computers and Graphics*, 10(2):161–170, 1986.

[24] Andrew S. Glassner. *Principles of Digital Image Synthesis.* Morgan Kaufmann, San Francisco, 1995.

[25] Gene Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[26] S. I. Granshaw. Bundle adjustment methods in engineering photogrammetry. *Photogrammetric Record*, 10(56):181–207, 1980.

[27] Paul R. Halmos. *Measure Theory.* Springer-Verlag, New York, 1974.

[28] Pat Hanrahan and Paul Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, pages 215–223, August 1990.

[29] Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 165–174, August 1993.

[30] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 175–186, July 1991.

[31] Paul S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986.

[32] Gerald C. Holst. *CCD Arrays, Cameras, and Displays.* SPIE Optical Engineering Press, Bellingham, Washington, 1996.

[33] Berthold K. P. Horn and Michael J. Brooks. *Shape from Shading.* MIT Press, Cambridge, Massachusetts, 1989.

[34] Katsushi Ikeuchi and Kosuke Sato. Determining reflectance properties of an object using range and brightness image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1139–1153, 1991.

[35] James T. Kajiya. The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, pages 143–150, August 1986.

[36] Konrad F. Karner, Heinz Mayer, and Michael Gervautz. An image based measurement system for anisotropic reflection. *Computer Graphics Forum (Eurographics '96 Proceedings)*, 15(3):119–128, August 1996.

[37] John K. Kawai, James S. Painter, and Michael F. Cohen. Radioptimization—goal based rendering. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 147–154, August 1993.

[38] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, NJ, 1974.

[39] Marc Levoy and Pat Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 31–42, August 1996.

[40] Song De Ma and Andre Gagalowicz. Determination of local coordinate systems for texture synthesis on 3-D surfaces. *Computers and Graphics*, 10(2):171–176, 1986.

[41] Stephen R. Marschner and Donald P. Greenberg. Inverse lighting for photography. In *Proceedings of the Fifth Color Imaging Conference.* IS&T and SID, November 1997.

[42] Saied Moezzi, Li-Cheng Tai, and Philippe Gerard. Virtual view generation for 3D digital video. *IEEE MultiMedia*, pages 18–26, January–March 1997.

[43] James R. Munkres. *Topology: A First Course.* Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[44] Shree K. Nayar, Katsushi Ikeuchi, and Takeo Kanade. Shape from interreflections. *International Journal of Machine Vision*, 6(3):173–195, 1991.

[45] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric considerations and nomenclature for reflectance. Monograph 161, National Bureau of Standards (US), October 1977.

[46] J. S. Nimeroff, E. Simoncelli, and J. Dorsey. Efficient re-rendering of naturally illuminated environments. In *Fifth Eurographics Workshop on Rendering*, pages 359–373, Darmstadt, Germany, June 1994.

[47] Eyal Ofek, Erez Shilat, Ari Rappoport, and Michael Werman. Multiresolution textures form image sequences. *IEEE Computer Graphics and Applications*, 17(2):18–29, 1997.

[48] A. P. Pentland. Finding the illuminant direction. *Journal of the Optical Society of America A*, 72:448–455, 1982.

[49] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, Cambridge, 1992.

[50] Marc Proesmans and Luc Van Gool. A sensor that extracts both 3D shape and surface texture. In *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 485–492. IEEE, 1996.

[51] Kari Pulli, Michael Cohen, Tom Duchamp, Hugues Hoppe, Linda Shapiro, and Werner Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. In *Rendering Techniques '97 (proceedings of the eighth Eurographics Rendering Workshop)*, pages 23–34. Springer-Verlag, June 1997.

[52] M. Rioux. Digital 3-D imaging: Theory and application. In *Videometrics III*, pages 2–15. SPIE, November 1994. Invited paper.

[53] Yoichi Sato and Katsushi Ikeuchi. Reflectance analysis for 3D computer graphics model generation. *Graphical Models and Image Processing*, 58(5):437–451, 1996.

[54] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 379–387, August 1997.

[55] Chris Schoeneman, Julie Dorsey, Brian Smits, James Arvo, and Donald Greenberg. Painting with light. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 143–146, August 1993.

[56] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, 1996.

[57] Patrick C. Teo, Eero P. Simoncelli, and David J. Heeger. Efficient linear re-rendering for interactive lighting design. Technical Report STAN-CS-TN-97-60, Stanford University, October 1997.

[58] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Machine Vision*, 9(2):137–154, 1992.

[59] K. E. Torrance and E. M. Sparrow. Off-specular peaks in the directional distribution of reflected thermal radiation. In *Transactions of the ASME*, pages 1–8, Chicago, Illinois, November 1965.

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

[61] Panagiotis Tsiotras, John L. Junkins, and Hanspeter Schaub. Higher order cayley transforms with applications to attitude representations. *Journal of Guidance, Control, and Dynamics*, 20(3):528–536, 1997.

[62] Greg Turk. Generating textures for arbitrary surfaces using reaction-diffusion. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 289–298, July 1991.

[63] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 311–318, July 1994.

[64] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 265–272, July 1992.

[65] D. Rod White, Peter Saunders, Stuart J. Bonsey, John van de Ven, and Hamish Edgar. Reflectometer for measuring the bidirectional reflectance of rough surfaces. *Applied Optics*, 37(16):3450–3454, 1998.

[66] Andrew Witkin and Michael Kass. Reaction-diffusion textures. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 299–308, July 1991.