

Multispectral BRDF Acquisition and Analysis for Computer Graphics

A Thesis

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Master of Science (Computer Science)

by

Colin Braley

May 2012

Abstract

The Bidirectional Reflectance Distribution Function, or BRDF, is used to model the reflectance properties of opaque materials. BRDFs are used in many fields, including applied physics, remote sensing, computer vision, and computer graphics. In computer graphics, artists specify BRDFs for each object in a virtual environment, and then render images of this environment. Traditionally, artists specify BRDFs through analytical models. However, tuning the parameters of these analytical models is time consuming, and often appropriate analytical BRDF models are not available. Increasingly, artists are turning to measured BRDF data to make this process easier. However, most existing measured BRDF data only has 3 color channels. Very little multispectral BRDF data is available, and the little data that is available has sparse angular sampling.

We present an image based BRDF measurement system capable of measuring multispectral BRDFs with dense angular sampling. We have measured several multispectral BRDFs, and we provide numerous plots, visualizations, and renderings of this data. We also present an analysis of several techniques used for calibrating BRDF measurement systems and algorithms for evaluating a BRDF from unstructured samples. Furthermore, we propose a new algorithm for extracting BRDF measurements from a set of images based on nonlinear optimization. This technique is simpler compared to techniques presented by previous authors.

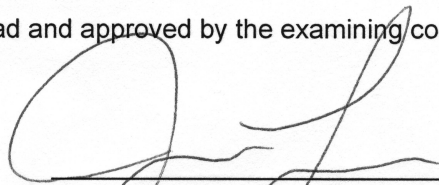
Additionally, we conducted a large scale statistical analysis of 59 million real world reflectance spectra. Using Principal Component Analysis(PCA), we validate the work of previous researchers and demonstrate that the space of real world spectral reflectances is spanned by a low-dimensional linear subspace. We also analyze this dataset using Non-Negative Matrix Factorization(NMF) and discuss the implications of our results for future multispectral BRDF measurement systems.

APPROVAL SHEET

The thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

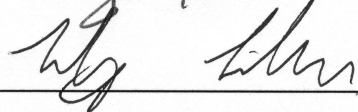
Colin Braley
AUTHOR

The thesis has been read and approved by the examining committee:

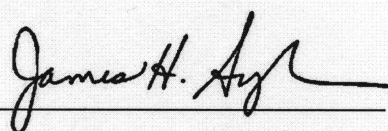

Advisor



Walter Martin



Accepted for the School of Engineering and Applied Science:



Dean, School of Engineering and Applied Science

May
2012

To my family and friends.

Acknowledgments

First, I would like to thank my adviser, Dr. Jason Lawrence for his guidance and support during my time in graduate school. I couldn't have asked for a better adviser.

Throughout my research I stumbled across numerous technical challenges and hurdles. Many people helped me work through these issues, and without them, this thesis would not be possible. I'd like thank Mike Holroyd, Michael Douglas, and Avinash Kalyanaraman for their help with numerous technical challenges.

Secondly, I would like to thank all of my friends in graduate school for making my time at UVA memorable and fun. In particular, I'd like to thank Avinash Kalyanaraman, Samee Zahur, Ben Rhodes, Panayiotis Steele, Yajun Wang, Ben Kreuter, and Rebecca Stewart.

Lastly, and most importantly, I'd like to thank my girlfriend Emily and my family for listening to me rant endlessly about BRDFs and rendering.

Contents

Contents	v
List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Introduction	1
1.2 Original Contributions	2
2 Background	4
2.1 The Physics of Light and Radiometry	4
2.1.1 Geometric Optics	4
2.1.2 The Physics of Light	5
2.1.3 Radiometric Quantities	5
2.1.4 Material Properties and the BRDF	6
2.1.5 Light Measurement Devices	8
2.1.6 Polarization	9
2.2 Color and Human Vision	10
2.2.1 Grassman’s Laws, The Linearity of Color, and Color Matching	11
2.2.2 The Color Matching Experiment: A Conceptual Overview	12
2.2.3 Details of the Color Matching Experiment and the CIE XYZ Color Space	14
2.3 Color Spaces and Display of Spectral Imagery	15
2.3.1 Linear Transforms Between Color Spaces	15
2.3.2 Other Display Considerations	16
3 Related Work	17
3.1 Analytical BRDF Models	17
3.2 BRDF Measurement Systems	17
4 Spectral Camera Calibration	19
4.1 Camera Calibration Goals	19
4.1.1 Mathematical Model	20
4.1.2 Spatial Calibration	22
4.1.3 Spectral Calibration	22
4.1.4 Spectral Reconstruction	25
4.1.5 Nyström Reconstruction	26
4.1.6 Multispectral Imaging Results and Comparison	26
4.1.7 Multispectral Relighting Results	28
5 BRDF Measurement	30
5.1 Measurement Algorithm Overview	30
5.1.1 Components of BRDF Measurement System	31
5.2 Converting HDR values to BRDF Measurements	32
5.2.1 Interpreting Pixel Values as BRDF Measurements	33

5.2.2	View Vector Determination	34
5.2.3	Surface Normal Determination	34
5.2.4	Light Vector Determination	34
5.2.5	Unstructured BRDF Data Results	35
5.3	BRDF Reconstruction from Scattered Data	36
5.3.1	Distance Metrics for Unstructured BRDF Samples	41
5.3.2	Reconstruction Methods for Unstructured BRDF Samples	41
5.4	Evaluation of BRDF Resampling	42
5.4.1	Torrance Sparrow BRDF Results	42
5.4.2	Ward BRDF Results	43
5.4.3	Parameter Choices for BRDF Reconstruction	44
6	BRDF Measurement Results	53
7	Multispectral Image Analysis	56
7.1	Introduction to Multispectral Imaging	56
7.2	A Database of Multispectral Images	57
7.3	Principal Component Analysis on Multispectral Images	58
7.3.1	Comparison with the Parkkinen Basis	60
7.4	Non-Negative Matrix Factorization on Multispectral Images	60
7.5	Implications of PCA and NMF Analysis for BRDF Measurement	63
8	Conclusion	64
	Appendix A Spectral Environment Map Generation	65

List of Tables

2.1	Table of radiometric units and associated notation	4
4.1	Table of mathematical notation for calibration	20

List of Figures

2.1	Lambert's cosine law	6
2.2	Angular parameterization of the BRDF in spherical coordinates.	7
2.3	Comparison of anisotropic and isotropic BRDF angular parameterization	9
2.4	The visible spectrum	10
2.5	CIE Color Matching Experiment	13
2.6	CIE XYZ Color Matching Functions(CMFs)	15
4.1	Camera calibration using integrating sphere.	22
4.2	Spectral calibration targets	23
4.3	Transmittance of Varispec multiplied by QE modelled as Dirac deltas	24
4.4	Transmittance of Varispec multiplied by QE modelled using a factored model	25
4.5	Screenshot of our spectral image viewer application	25
4.6	Comparison of spectral calibration techniques using red target	27
4.7	Comparison of spectral calibration techniques using green target	27
4.8	Comparison of spectral calibration techniques using blue target	27
4.9	Comparison of spectral calibration techniques using yellow target	28
4.10	Comparison of multispectral and RGB relighting results.	29
5.1	Diagram of image based BRDF measurement system using spherical samples.	31
5.2	Photographs of BRDF measurement system.	32
5.3	Visualization of non-linear minimization used for circle detection.	33
5.4	The incidence plane of a BRDF	35
5.5	Incidence plane BRDF plots of a glossy red Christmas ornament	37
5.6	Incidence plane BRDF plots of highly specular red plastic	38
5.7	Incidence plane BRDF plots of a blue racquetball.	39
5.8	Ground truth render for Torrance-Sparrow BRDF.	43
5.9	Comparison of reconstruction parameters on synthetic Torrance Sparrow BRDF data($k = 1$)	44
5.10	Comparison of reconstruction parameters on synthetic Torrance Sparrow BRDF data($k = 16$)	46
5.11	Comparison of reconstruction parameters on synthetic Torrance Sparrow BRDF data($k = 64$)	47
5.12	Comparison of reconstruction parameters on synthetic Torrance Sparrow BRDF data($k = 256$)	48
5.13	Ground truth render for isotropic Ward BRDF.	49
5.14	Comparison of reconstruction parameters on synthetic isotropic Ward BRDF data($k = 1$)	49
5.15	Comparison of reconstruction parameters on synthetic isotropic Ward BRDF data($k = 16$)	50
5.16	Comparison of reconstruction parameters on synthetic isotropic Ward BRDF data($k = 16$)	51
5.17	Incidence plane plot of specular red plastic unstructured data with binned data overlaid.	52
6.1	Renderings of spheres with our measured spectral BRDFs.	54
6.2	Renderings of several multispectral BRDFs applied to the Stanford Dragon Model	55
7.1	Example multispectral image converted to sRGB.	57
7.2	First 8 principal components obtained from PCA of 59 million real world reflectance spectra.	59
7.3	Eigenvalue magnitude plot from PCA	59
7.4	Multispectral image projected onto principal components	60

7.5	Comparison with the Parkkeinnen basis	61
7.6	Results from non-negative matrix factorization(NMF)	62
7.7	Per-pixel L^2 error from NMF based projection	62

Chapter 1

Introduction

1.1 Introduction

One of the primary goals of computer graphics is to synthesize, or render, photorealistic imagery of virtual environments. These environments are instead created by 3D artists. Artists create a *scene description*, and then rendering software is used to generate photorealistic images of this scene. Scene descriptions include placement of light sources, the positioning of a virtual camera, the geometric content of the scene, and the material properties of the objects in the scene.

Generating this scene description is challenging, and artists require tools to specify how a virtual world should look. Artists have some idea regarding what visual they would like to create. However, the process of conveying this via the scene description is challenging. One particularly difficult aspect of the scene to specify are *material properties*. There are many varieties of common materials that interact with light in different ways. Some materials scatter light below the surface, and exhibit what is known as subsurface scattering. Other materials exhibit varying degrees of transparency. Opaque materials scatter light outward from the surface. The reflectance of an opaque ¹ material is characterized by the *Bidirectional Reflectance Distribution Function*, or the *BRDF*. The BRDF specifies both what is colloquially thought of as the materials color, as well as its reflectance (shininess, glossiness, matte-ness). Mathematically, how much light a material reflects is both a function of the incoming direction of light relative to the surface normal, the outgoing direction, and the wavelength of light.

Artists need to accurately specify BRDFs in order to generate realistic imagery. Traditionally, artists have used analytical models to specify BRDFs. Analytical BRDF models are mathematical functions with user

¹By opaque, we mean a material that does not exhibit subsurface scattering. Physically, all non-metals exhibit some small amount of subsurface scattering[1]. However, for many materials this scattering is negligible to the point where it can be ignored. Therefore, it is useful to measure BRDFs of non-metals.

settable parameters. By choosing different models, and by altering parameter values, artists are mimic the appearance of real world materials. However, analytical BRDF models have several drawbacks. First, it is not always clear which model, if any, suits a particular type of material an artist wishes to create. For certain classes of materials, no analytical models have been developed. Secondly, the process of tuning parameters for an analytical BRDF model can be laborious. Therefore, the computer graphics community has increasingly turned towards measuring BRDFs directly and then using this data in the rendering software.

The BRDF is a function of 4 angular variables(in the anisotropic case) and 1 wavelength variable. As a 5 dimensional function, sampling the BRDF sequentially at a series of points in this 5D domain is very slow. Therefore, the graphics community has turned to *image based BRDF capture* to speed up the measurement process[2][3][4]. Image based BRDF capture systems use cameras as the measurement device, and interpret each pixel as a separate BRDF measurement. This enables parallelism in the measurment process. Image based BRDF measurement systems have also enabled other interesting applications. For example, Matusik et. al. created a database of over 100 measured BRDFs using an image based capture device[4]. Analysis of this large database allowed new algorithms and techniques to be developed for alternative sparse BRDF measurement techniques, compression of BRDF data, and editing of measured data.

Due to the tristimulus property of human vision, 3 numbers are sufficient to describe any color stimulus under normal viewing conditions. Therefore, the final goal of a rendering system is typically the creation of an accurate RGB image of the scene in question. However, for maximum accuracy the rendering software should not perform intermediate computations using RGB colors! Intermediate computations should be done with a spectral representation of color, and conversion to RGB should be a final step. However, spectral rendering is underutilized because of a lack of spectral reflectance data and analytical spectral BRDF models[5].

1.2 Original Contributions

In this thesis, we describe a new multispectral image based BRDF measurement system. This system has produced some of the first multispectral BRDFs with both dense angular and spectral sampling. Our measurement device has several unique aspects to it that make it particularly useful. We employ a new, novel calibration scheme for determining per-pixel surface normals, view vectors, and light vectors. Our approach is based on taking silhouette images of our measurement targets and using nonlinear optimization to determine a scene that produces the correct silhouette. Given a reasonable initial guess, our process converges quickly and to a global minimum, thus producing correct results. Compared to the approach of Matusik et al.[6], our approach is simpler and less costly. We present initial results using both measured and synthetic data to validate our BRDF measurement system.

Secondly, we present a thorough analysis of the spectral calibration process. Spectral calibration is necessary in order to extract meaningful radiometric measurements from cameras. While the majority of this calibration is not new, it has not been evaluated as applied to BRDF measurement.

Finally, we conduct a statistical analysis of a large database of multispectral images previously acquired by Chakrabarti and Zickler[7]. We perform Principal Component Analysis(PCA) on this large 13 Gigabyte dataset and confirm previous results of Parkkinen et al.[8]. We also analyze this large dataset using Non-Negative Matrix Factorization(NMF)[9]. We present several areas of future work in which these analyses could be applied to new BRDF measurement systems.

Chapter 2

Background

2.1 The Physics of Light and Radiometry

To understand both material measurement systems and rendering algorithms we must have a systematic way of measuring light. *Radiometry* is the branch of physics that studies how electromagnetic radiation(light) propagates in an environment. A full, detailed review of radiometry is outside the scope of this document. For a well written introduction aimed at a graphics audience, we recommend the book chapter by Hanrahan[10]. For a mathematical and thorough treatment of radiometry, including pointers to relevant historical literature, we recommend Veach’s thesis[11]. For those not interested in reviewing radiometry, a table of the notation we use for radiometric units throughout this thesis is shown in table 2.1.

Radiometric Units		
Symbol	Radiometric Quantity	Units
Φ	Flux(radiant power)	W
E	Irradiance	$\frac{W}{m^2}$
L	Radiance	$\frac{W}{m^2 sr}$
E_λ	Spectral Irradiance	$\frac{W}{m^3}$
L_λ	Spectral Radiance	$\frac{W}{m^3 sr}$

Table 2.1: Table of radiometric units and associated notation. Note that for spectral quantities the $\frac{1}{m^2}$ term in the units becomes $\frac{1}{m^3}$. This is because $\frac{1}{m^2 nm} = \frac{1}{m^3 10^{-9}}$.

2.1.1 Geometric Optics

Light has properties of both particles and waves, through what is referred to as the *wave-particle duality* of light. To describe certain observable phenomenon, like diffraction, we must consider the wave nature of light. In general, wave based models of light(such as Maxwell’s equations) have not found significant usage in

computer graphics. Wave based effects are normally visible when the wavelength of the light is comparable in size to the geometry being rendered. Typically, computer graphics is not concerned with rendering such small scale phenomenon. Instead, in computer graphics we typically use models based on *geometric optics*. Geometric optics is simpler than a wave based model, and still accounts for a large variety of observable phenomenon. Under the geometric optics model, several simplifying assumptions are made that are not true in the general case, but are good approximations of reality for many scenes we are concerned with.

The first simplifying assumption is that light travels in straight lines. In general, this is not true. For example, we assume that there is no participating media(fog,dust) present to scatter light. Furthermore, we assume that the index of refraction of air is constant spatially¹. Other effects, like diffraction, violate the assumption that light travels in straight lines. The next simplifying assumption is that we ignore the temporal aspect of light. In reality, light takes time to travel to any location. However, we are concerned with generating an image of the steady-state distribution of light, and therefore we ignore these temporal aspects.

2.1.2 The Physics of Light

Being a form of electromagnetic radiation, light is a wave formed by both an electric and magnetic field. These two fields are perpendicular to the direction of travel. The frequency of this wave determines the wavelength of the light. Light of different wavelengths is perceived by humans as different colors. Every radiometric quantity we discuss in this section also has a *spectral* form, which is the quantity in question per nanometer. However, in the interest of simplifying the discussion we consider only monochromatic(single wavelength) light in this section and defer the discussion of color and spectral quantities until section 2.2.

2.1.3 Radiometric Quantities

The most fundamental radiometric quantity is flux, or power. Flux is commonly denoted by the symbol Φ , and is expressed in Watts($\frac{J}{s}$,Joules per second). Flux describes only how much energy flows through a surface per unit time. The size of the surface is irrelevant to the measurement of flux.

Irradiance, commonly denoted by E , is the radiant power arriving at a surface, per unit surface area. A corresponding quantity, *radiant exitance*, describes the energy leaving a surface, and is denoted by M . Being differential quantities, we express both irradiance and radiant exitance as $\frac{d\Phi}{dA}$. The units of both quantities are $\frac{W}{m^2}$ (Watts per square meter).

¹If the index of refraction of air varies smoothly along a ray's path, mirage-like effects occur.

The most important radiometric quantity in computer graphics is radiance, denoted by L . Radiance is fundamental in that all other radiometric quantities can be derived from it. Radiance is defined as flux per unit projected area, per unit solid angle. Its units are $\frac{\text{W}}{\text{m}^2 \text{sr}}$. Note that sr stands for steradians, the measure of solid angle. The radiance at a surface is given by the following equation[12]:

$$L = \frac{d^2\Phi}{d\omega dA^\perp} = \frac{d^2\Phi}{d\omega dA \cos(\theta)} \quad (2.1)$$

The notation A^\perp denotes projected area. Intuition for why this term is included can come from Lambert’s cosine law. As a beam of light is shines nearly parallel to a surface(a grazing angle), the light beam is “smeared” over a larger physical area. This is shown in figure 2.1.

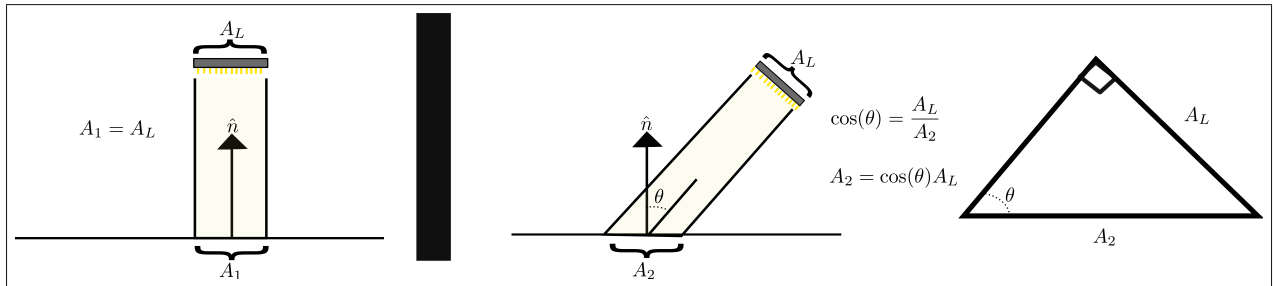


Figure 2.1: Illustration of Lambert’s law. Inspired by figure 5.7 in *Physically Based Rendering*, by Pharr and Humphreys[13]. We see as the angle θ approaches 90 degrees, the area of A_2 increases.

The definition of radiance initially seems convoluted and unnecessary. However, when defined as shown above radiance possesses several properties that make it enormously useful. First, radiance has the *property of invariance*[12]. The radiance along a straight line from point a to b is the same as the radiance from b to a .

Secondly, the human eye and most camera sensors are sensitive to spectral radiance L_λ , or radiance per unit wavelength. Therefore, rendering systems that are simulating either the human eye or a camera aim to simulate radiance.

2.1.4 Material Properties and the BRDF

When light hits a surface it can be absorbed, transmitted, or reflected. In the case of absorption, light is dissipated as heat in the surface. For example, asphalt absorbs much of the light across the visible spectrum, and this is why asphalt feels hotter than white paint which reflects light across the spectrum. In the case of transmission, light passes through the surface. We do not consider transmission of light through surfaces in this thesis. We limit our discussion to the BRDFs, which are relevant for opaque surfaces.

Reflection from an opaque surface can be described by the Bidirectional Reflectance Distribution Function(BRDF). The BRDF was first described by Nicodemus[14]. The BRDF describes, as a function of incoming angle, outgoing angle, and wavelength, how a particular material reflects light. In this thesis we ignore phosphorescence and fluorescence. Phosphorescent materials absorb light and then re-emit it some small amount of time later. Fluorescent materials take incident light at a wavelength λ_i and reflect it at another wavelength $\lambda_o | \lambda_o > \lambda_i$. Fluorescence accounts for why a white shirt appears purple under a black light. This is because the UV light from the black light is being reflected(technically, re-radiated) as low-wavelength(purple) visible light. Fluorescent BRDFs were recently measured for the first time in computer graphics by Hullin et al[15].

The Bidirectional Reflectance Distribution Function(BRDF)

Typically, the BRDF is written as a function of two unit vectors and of wavelength, as shown in equation 2.2. These two vectors encode the incident and outgoing directions of reflectance relative to the surface normal.

$$f_r(\hat{\omega}_i, \hat{\omega}_o, \lambda) \quad (2.2)$$

Here the vector $\hat{\omega}_i$ represents the direction of incident light, whereas the vector $\hat{\omega}_o$ represents outgoing light. The BRDF measures how much light is scattered in direction $\hat{\omega}_o$ when arriving in direction $\hat{\omega}_i$. By convention, both vectors point outward from the surface and are unit length. If we consider each vector in spherical coordinates, we can write the angular portion of the BRDF as a function of 4 variables. We refer to a vector's angle from the surface normal as the *elevation angle*, given by either θ_i or θ_o for the incident and outgoing vector, respectively. The angle relative to some tangent vector \hat{t} is called the *azimuth angle*, and is denoted by ϕ_i and ϕ_o . The angular parameterization of the BRDF is shown in figure 2.2.

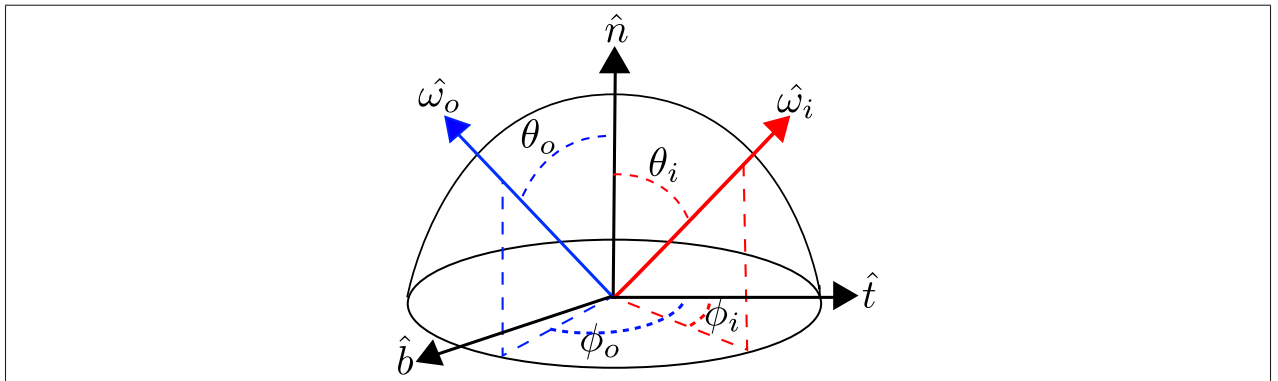


Figure 2.2: Angular parameterization of the BRDF in spherical coordinates.

Properties of the BRDF

There are several properties common to all BRDFs. The first property is related to conservation of energy².

$$\forall \hat{\omega}_i \in \Omega \text{ and } \forall \lambda \in \mathbb{R}^+ : \int_{\Omega} f_r(\hat{\omega}_i, \hat{\omega}_o, \lambda) \cos(\theta_o) d\hat{\omega}_o \leq 1 \quad (2.3)$$

In the above equation Ω represents the hemisphere around the surface normal. This integral must be less than 1 because the surface itself can't create energy, so less light must be reflected than is incident. The $\cos(\theta_o)$ term in this equation accounts for “foreshortening,” and is related to Lambert’s law. See section 2.1 for more details on the origin of this term.

A second, more interesting property of the BRDF is *Helmholtz reciprocity*³. Helmholtz reciprocity states that the BRDF must remain unchanged if we swap the incident and outgoing light directions.

$$f_r(\hat{\omega}_i, \hat{\omega}_o, \lambda) = f_r(\hat{\omega}_o, \hat{\omega}_i, \lambda) \quad (2.4)$$

Isotropic vs Anisotropic BRDFs

Many BRDFs possess the property of *isotropy*. Isotropic BRDFs, as compared with anisotropic BRDFs, are symmetric about rotation around the surface normal. Isotropic BRDFs have no “grain direction.” Examples of anisotropic BRDFs include things like brushed velvet, brushed aluminum, or some milled metal materials. Isotropic BRDFs can be represented with 3 angular degree of freedom, as opposed to the 4 that are required for anisotropic materials. Instead of accounting for both ϕ_i and ϕ_o , we can instead consider just $\phi_{\text{diff}} = \phi_i - \phi_o$, owing to the BRDFs symmetry about the normal. The anisotropic and isotropic parameterizations are compared in figure 2.3.

2.1.5 Light Measurement Devices

There are many physical devices for measuring radiometric quantities. For our purposes, how these devices work can be treated largely as a black box. We are only concerned with what these devices measure and their mathematical properties. Temporarily ignoring wavelength dependence, we state that *sensors measure radiance*, if radiance is constant across the field of view of the sensor. Fortunately, the field of view for individual sensors (usually pixels) is extremely small. Consider an individual pixel on a camera CCD. This

²The property stated in equation 2.3 does not hold for fluorescent materials, since light can be re-radiated from a shorter wavelength to a longer one. See Hullin et. al[15]. for more details.

³Interestingly, Veach notes[11] that Helmholtz himself never actually stated anything related to what is now known as Helmholtz reciprocity. Furthermore, Veach also notes 2 “exotic” situations that can cause Helmholtz reciprocity to breakdown. The situations (involving either participating media or magnetic fields) are thankfully not things we commonly wish to render.

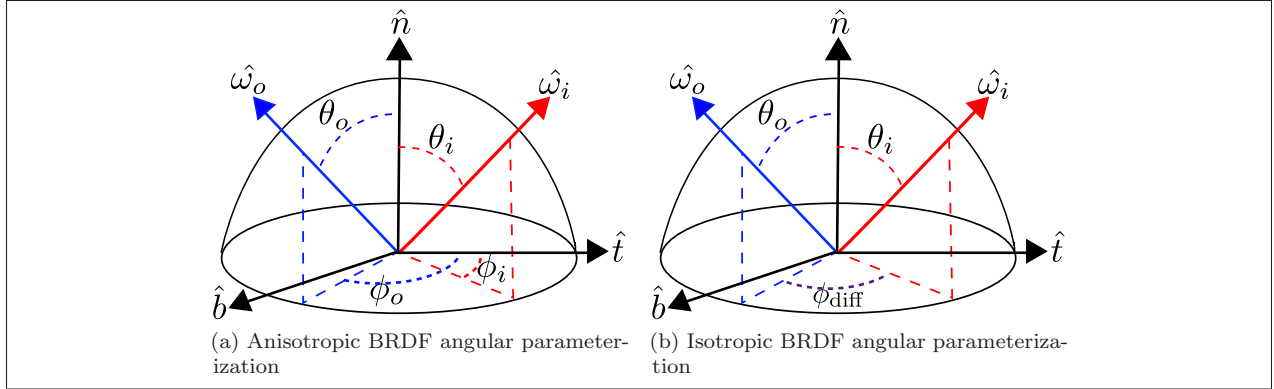


Figure 2.3: Comparison of anisotropic and isotropic BRDF angular parameterization

pixel takes of some small area $a \text{ m}^2$. Ignoring the dynamic range of the sensor, the value read off at this pixel, V , is proportional to the integral of the flux falling upon the sensor.

$$V \propto \int_a \int_{\Omega} L \cos(\theta) d\omega da \quad (2.5)$$

When a photon of a particular wavelength strikes a pixel, there is a probability that it will bounce off and not be recorded. The probability that a photon of a particular wavelength will be recorded is given by the *quantum efficiency function*, or QE. This is sometimes also called the *spectral sensitivity* of the sensor. Quantum efficiency functions are functions Q of the form $Q : \mathbb{R}^+ \rightarrow [0, 1]$. For example, $Q(420) = .1$ would indicate that for every 10 photons of wavelength 420 nm, 9 of them bounce off and are not recorded. Camera's with multiple color channels will have different quantum efficiency functions for each channel. For a camera with k channels, there will be k different quantum efficiency functions⁴.

2.1.6 Polarization

In computer graphics, we do not often consider the polarization of light. However, polarization can be important when measuring BRDFs. Polarization refers to the orientation of the magnetic field and electric field vectors relative to the direction of propagation of a ray of light. When the orientation of these vectors is random, light is said to be *unpolarized*, or *incoherent*. Most rendering systems only simulate unpolarized light. Types of polarization include *linear polarization*, *circular polarization*, and *elliptical polarization*. A general discussion of the full nature of polarized light is beyond the scope of this document. However, reflection from arbitrary BRDFs (particularly specular reflection) can linearly polarize incident incoherent light. Typically this is not an issue, as normal CCD and CMOS image sensors largely work the same

⁴Some color imaging systems operate with a monochrome CCD and thus have 1 quantum efficiency function. Color imaging is achieved by swapping various colored transmission filters in front of the CCD. While the mechanism here is different, mathematically this is equivalent to multiple quantum efficiency functions.

regardless of polarization. However, Liquid Crystal Tunable Filters(LCTFs) act as linear polarizers. This can be an issue in multispectral BRDF measurement systems, as discussed in section ??.

2.2 Color and Human Vision

The human eye is sensitive to electromagnetic radiation(light) from the wavelengths 380 nm to 780nm. Below 380nm is ultraviolet light(UV), and above 780nm exists infrared light(IR). IR and UV light are invisible to humans. Light near the upper and lower boundary of the visible spectrum is deemed near-UV and near-IR light, respectively. Light emitted in different regions of the visible spectrum produce different color sensations in human observers. The familiar color rainbow starts at 380nm with violet light, and progresses through blue, green, yellow and red. A visualization of the visible spectrum is shown in figure 2.4.



Figure 2.4: The distribution of light across the visible spectrum. At each wavelength λ' we show the color of monochromatic light with wavelength λ' .

Light arriving at the human eye along a particular direction possesses what is called a *spectral power distribution*(SPD), or a spectrum. This spectrum describes the amount of energy present at each wavelength. Different light spectra produce different color sensations when viewed by a human observer. The study of how humans perceive color relied on radiometry, psychology, linear algebra, and biology. The study of color encompasses a field of study in and of itself: color science. A full review of color science is far beyond the scope of this thesis. We recommend readers to the book *Radiosity and Realistic Image Synthesis*[10] for an overview of color for a graphics practitioner. For an encyclopedic, exhaustive review of early color science, including a detailed description of the color matching experiments, we recommend[16]. Here we describe the fundamentals of color science so that our research contributions can be appreciated in a larger context.

There are an infinite number of possible light spectra. However, light can(under most conditions) be described by 3 numbers. This remarkable fact is known as the tristimulus property of human vision. The tristimulus property of human vision has a biological basis. Human eyes possess two main types of light detectors: rods and cones. Rods are only active in low light situations, and cones are only active in “normal” light situations. The situation when the rods are active is referred to as scotopic vision, whereas the cones being active is referred to as photopic vision. An interesting effect called the Purkinje shift occurs as we transition between photopic and scotopic vision. This effect causes colors to shift to be more purple due to

the interaction of the rods and the cones. Situations in which both rods and cones are active is referred to as Mesopic vision. However, in graphics we are typically concerned only with photopic vision, since that is when most images are viewed, and in daily life photopic vision is the common case. Therefore, for the rest of this thesis we will discuss only photopic vision. Note that, in general, the laws of linearity discussed later do not hold for scotopic vision[16]!

2.2.1 Grassman's Laws, The Linearity of Color, and Color Matching

A fundamental concept in color science is color matching. Color matching consists of having a human observer look at a fixed light spectrum called the “test light”. The user then adjusts the brightness of another set of lights, the “primaries,” until the mixture of colors produced by the primaries appears the same as the test light. We denote our test light by the $T(\lambda)$. We denote our N primaries by $P_1(\lambda), \dots, P_N(\lambda)$. The operation of altering the brightness of a light is given by multiplying the light's spectrum by a scalar α_i for light $P_i(\lambda)$. This scalar is called a “matching weight.” When two lighting configurations look the same to the human observer, we call this scenario a “color match.” Mathematically, if spectra $A(\lambda)$ and $B(\lambda)$ match we denote this by $A(\lambda) \sim B(\lambda)$. Therefore, the situation described above of matching our test light $T(\lambda)$ to our N weighted primaries $P_1(\lambda), \dots, P_N(\lambda)$ is given by:

$$T(\lambda) \sim \alpha_1 P_1(\lambda) + \dots + \alpha_N P_N(\lambda) \quad (2.6)$$

Color science is based off of the fundamental fact that color matching is linear⁵ This was first noted as early as 1853 by Hermann Grassmann[16]. Assume we are dealing with two lights $A(\lambda)$ and $B(\lambda)$, and N primaries $P_1(\lambda), \dots, P_N(\lambda)$. Also assume that light $A(\lambda)$ is matched by $\alpha_1 P_1(\lambda) + \dots + \alpha_N P_N(\lambda)$ and light $B(\lambda)$ is matched by $\beta_1 P_1(\lambda) + \dots + \beta_N P_N(\lambda)$. Grassman's laws then state that the light $A(\lambda) + B(\lambda)$ is matched according to the following equation:

$$A(\lambda) + B(\lambda) \sim (\alpha_1 + \beta_1) P_1(\lambda) + \dots + (\alpha_N + \beta_N) P_N(\lambda) \quad (2.7)$$

Grassman's laws are profound in terms of what they imply mathematically. Due to the linearity of color matching, we can apply the tools of linear algebra to color science. The set of all light spectral power distributions form a vector space.

It is also important to note that the linearity of light holds for light measurement as well as for color matching. We can compare two light spectra radiometrically by measuring them with a spectroradiometer.

⁵Note that in general the linearity of color matching only holds in photopic conditions.

When measuring two lights, linearity holds as well:

$$\alpha(A(\lambda) + B(\lambda)) = \alpha A(\lambda) + \alpha B(\lambda) \quad (2.8)$$

2.2.2 The Color Matching Experiment: A Conceptual Overview

Note that in this section we describe the color matching experiment only conceptually. For practical purposes the actual color matching experiment was conducted somewhat differently. For basic information on how the actual experiment took place, see section 2.2.3 and refer to the text “Color Science[16].”

Any SPD $T(\lambda)$ can be matched by 3 linearly independent primaries P_1, P_2, P_3 . The primaries are linearly independent if the following equation is satisfied for some a, b, c such that all a, b, c are not zero:

$$\exists a, b, c \in \mathbb{R} \text{ such that } aP_1(\lambda) + bP_2(\lambda) + cP_3(\lambda) = S_0(\lambda) \quad (2.9)$$

where $S_0(\lambda)$ denotes the “zero spectrum” $S_0(\lambda) = 0$. The “zero spectrum” is analogous to the zero vector in a vector space.

All light spectra can be matched by any 3 independent primaries. This is known as the “principal of trichromancy.” Mathematically, this can be stated as follows:

$$\forall L(\lambda) \exists a, b, c \in \mathbb{R} \text{ such that } L(\lambda) \sim aP_1(\lambda) + bP_2(\lambda) + cP_3(\lambda) \quad (2.10)$$

The principal of trichromancy allows us to create colorspace like sRGB and XYZ. When we wish to describe the color of a SPD to a display device, 3 numbers is sufficient. However, require a procedure to determine, for any arbitrary spectrum $L(\lambda)$, what the matching weights a, b, c are for an arbitrary set of primaries. Since there are an infinite number of SPDs, we can’t simply experimentally determine all of them. A fundamental experiment conducted in 1931 by the CIE; the color matching experiment, derived data that allows us to compute matching weights for any SPD. The experimental setup consists of a human observer looking at a surface with flat spectral reflectance (such as Spectralon). This surface is divided into two sides. On the left side, a test light source with spectrum $T(\lambda)$ is illuminating the surface. On the right side, a linear combination of the three primaries illuminates the surface. This linear combination is given by $aP_1(\lambda) + bP_2(\lambda) + cP_3(\lambda)$. The human subject has access to a control box and they can adjust the matching weights a, b and c interactively. The user adjusts the weights until the following condition is satisfied:

$$aP_1(\lambda) + bP_2(\lambda) + cP_3(\lambda) \sim T(\lambda) \quad (2.11)$$

This setup is shown from a top-down view in figure 2.5. This experimental setup allows us to generate the matching weights for any test light T .

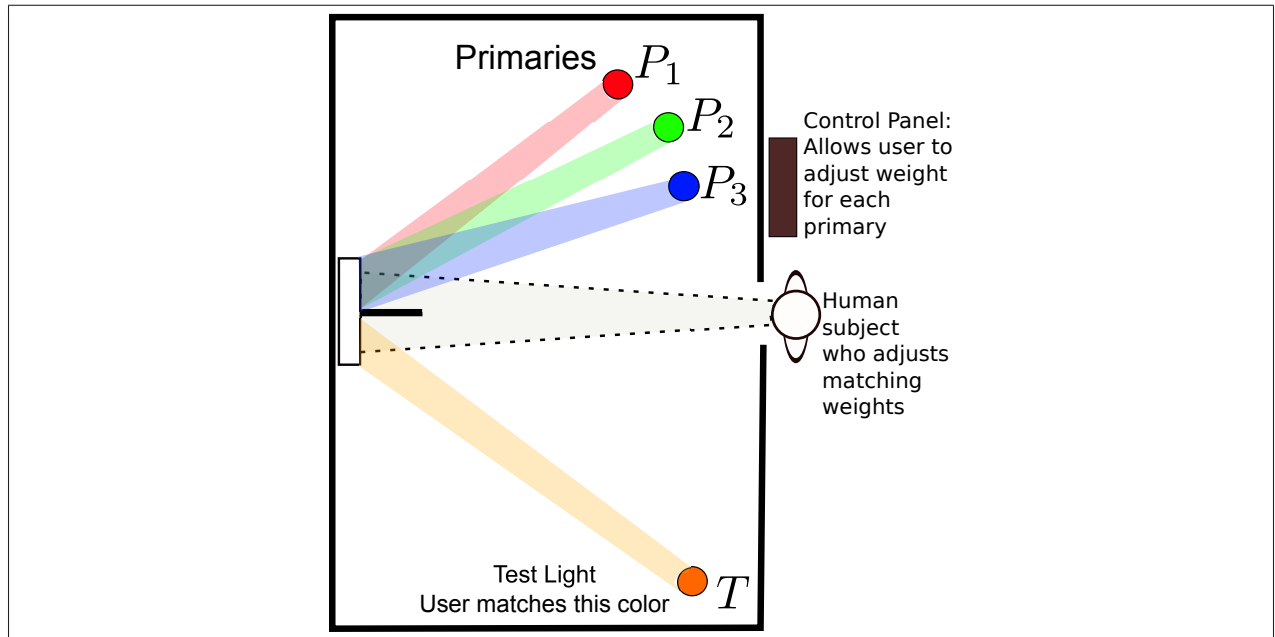


Figure 2.5: CIE Color Matching Experiment

If we conduct this experiment for a series of monochromatic (single wavelength) light sources T , we can then use the resulting data to compute the matching weights for any arbitrary light source. This is due to the linearity of color matching. Mathematically, we denote a monochromatic light emitting only a single unit of spectral radiance at $\lambda = w$ by a shifted Dirac delta function: $T_w(\lambda) = \delta(w - \lambda)$.

Assume we have collected data for a series of monochromatic light sources. These monochromatic sources emit light at a series of discrete wavelengths w_1, \dots, w_N . In the actual color matching experiment $w_1 = 380\text{nm}$, and $w_N = 780\text{nm}$. We denote the matching weights for the i th monochromatic light source by x_i, y_i , and z_i , where these correspond to primaries P_1, P_2 , and P_3 , respectively.

Any arbitrary SPD can be constructed via a linear combination of these monochromatic sources. Therefore, for any arbitrary SPD $T'(\lambda)$ we can generate our matching weights X, Y, Z via the following summations:

$$\begin{aligned} X &= \sum_{i=1}^N T'(w_i) x_i \\ Y &= \sum_{i=1}^N T'(w_i) y_i \\ Z &= \sum_{i=1}^N T'(w_i) z_i \end{aligned} \tag{2.12}$$

In the limit, as N gets very large and samples get closer and closer together we can think of these summations as integrals. We then have a function $x(\lambda)$ which gives us the matching weight for primary $P_1(\lambda)$, and similar functions $y(\lambda)$ and $z(\lambda)$ for primaries P_2 and P_3 . Our matching equations then become:

$$\begin{aligned} X &= \int_{\lambda=w_1}^{\lambda=w_N} T'(\lambda) \bar{x}(\lambda) d\lambda \\ Y &= \int_{\lambda=w_1}^{\lambda=w_N} T'(\lambda) \bar{y}(\lambda) d\lambda \\ Z &= \int_{\lambda=w_1}^{\lambda=w_N} T'(\lambda) \bar{z}(\lambda) d\lambda \end{aligned} \tag{2.13}$$

These functions $x(\lambda)$, $y(\lambda)$, and $z(\lambda)$ are known as the color matching functions. They have been computed by numerous different experimenters with various experimental setups. Typically, these functions are stored tabulated at 1nm or 5nm intervals, and the definite integrals shown in equation 2.13 are turned into Riemann summations. With this tabulated data, we can compute the matching weights for any arbitrary SPD.

2.2.3 Details of the Color Matching Experiment and the CIE XYZ Color Space

The observation conditions in the color matching experiment have an effect on the results. The human observer typically observes a circular patch of color, but this patch does not fill up the persons full field of view. As the angle subtended by the human observer's eye changes, different results are obtained since the distribution of rods and cones is not uniform across the entire eye. The two main variants of the color matching functions are 2 degree and 10 degree. The 2 degree CMFs are designed such that the field of view only covers the fovea of the eye [16]. The two degree CMFs are better to use when designing imagery that will be viewed on a small device (like a monitor) from a distance. The 10 degree CMFs make more sense for designing something like imagery shown in a movie.

The astute reader may have noticed a mathematical issue with the color matching experiment described above. As simply stated, the color matching experiment does not allow for negative matching weights. Without negative weights, we can't form an arbitrary linear combination of the primary lights. Mathematically having a primary with a negative weight is equivalent to adding light to the target SPD. This is how negative weights can be implemented in practice. It turns out that either negative weights or negative primaries (sometimes called imaginary primaries) are necessary in practice to allow a match for all colors.

A final issue in the color matching experiment is the choice of primary sources $P_1(\lambda)$, $P_2(\lambda)$, and $P_3(\lambda)$. These SPDs must be linearly independent. However, using we can take data with any arbitrary linearly independent primaries and then transform the results into a form as if the data was recorded with different

primaries. This change of basis approach is useful in that we can transform the CMFs into a form where we have negative values primaries. The change of basis matrix will simply be a 3 by 3 matrix with each element being the inner product of a pair of an “old” primary with a “new primary.” This process is described more in section 2.3. This was done with the CIE XYZ color space, and allows the matching functions to be positive valued. This also has the benefit of allowing whoever is conducting the color matching experiment to use whatever primary sources are most convenient. The matching functions for the XYZ color space are shown below in figure 2.6.

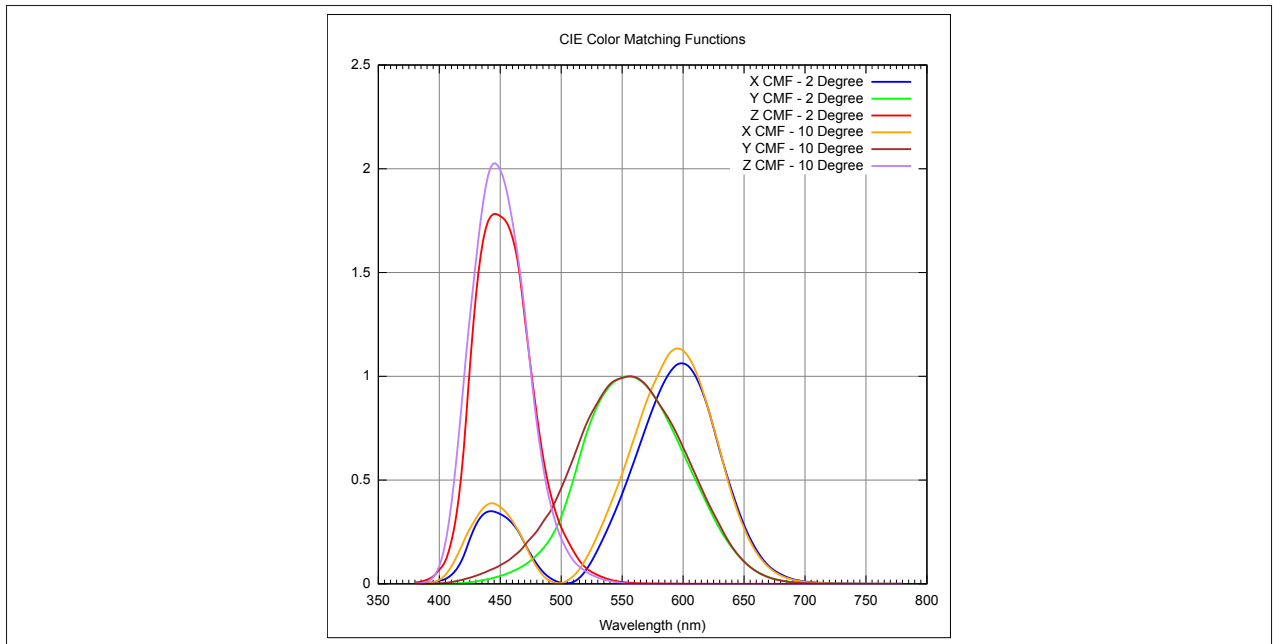


Figure 2.6: CIE XYZ Color Matching Functions(CMFs)

2.3 Color Spaces and Display of Spectral Imagery

A linear color space is defined by 3 known matching primaries. Different color spaces use different matching primaries. When working with a particular color space, we identify colors by 3 numbers that specify its coordinates in that three dimensional space.

2.3.1 Linear Transforms Between Color Spaces

Given an arbitrary SPD $S(\lambda)$ we often wish to display this color on a display device. Typically, this is done as follows. First, we compute the matching weights X, Y, Z using equation 2.13. This tells us what linear combination of the CIE XYZ primaries could be used to display our spectrum. However, our display will not match up with the XYZ primaries. Our display will have its own primary spectra. For a computer

LCD monitor, these corresponds to the SPD of the red, green, and blue phosphors in the display device. As described above, due to Grassman's laws, a linear transform can be used to transform our previously computed XYZ matching weights into matching weights for our display device. These display device matching weights tell us how what three tuple of numbers will allow our display to best approximate this color. Consider a color space defined by primaries $A(\lambda), B(\lambda), C(\lambda)$. We have already computed matching weights X, Y, Z using our tabulated XYZ CMFs. We now wish to compute matching weights a, b, c for our new color space. The linear transform required to do so is given by the following:

$$\begin{bmatrix} \int_{\Lambda} \bar{x}(\lambda)A(\lambda)d\lambda & \int_{\Lambda} \bar{y}(\lambda)A(\lambda)d\lambda & \int_{\Lambda} \bar{z}(\lambda)A(\lambda)d\lambda \\ \int_{\Lambda} \bar{x}(\lambda)B(\lambda)d\lambda & \int_{\Lambda} \bar{y}(\lambda)B(\lambda)d\lambda & \int_{\Lambda} \bar{z}(\lambda)B(\lambda)d\lambda \\ \int_{\Lambda} \bar{x}(\lambda)C(\lambda)d\lambda & \int_{\Lambda} \bar{y}(\lambda)C(\lambda)d\lambda & \int_{\Lambda} \bar{z}(\lambda)C(\lambda)d\lambda \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2.14)$$

Unfortunately, we usually do not know the primaries of our display device. Common approximations of display device primaries include those in the sRGB and Adobe RGB color spaces. In these cases, the XYZ to RGB matrices are available from a reference, and we do not need to compute this matrix ourselves.

2.3.2 Other Display Considerations

The display of computer generated images is complex topic in and of itself. A full discussion of this topic is outside the scope of this thesis. Therefore, we give a brief description of what needs to be considered, provide pointers to relevant literature, and describe how each issue is handled in our system. Issues discussed include tone mapping, gamma correction, and gamut mapping.

When converting a spectral image to an RGB color space for display we can get a large range of values for each color component. However, display devices typically only allow 8-bit values. Therefore, we must compress the dynamic range of our HDR image down to 8 bits. This process is called tone mapping.

Another issue is gamma correction. In general, when we double a value passed to a display device, the resulting radiance does not double. Rather, they obey a non-linear relationship. This can be corrected for via gamma correction. We refer the interested reader to Hall's book for information on gamma correction and other display considerations [17].

An additional issue is that different display devices have different gamuts. A color gamut describes the range of colors displayable by a particular device. Gamut mapping refers to how to deal with SPDs whose colors are not displayable(out of gamut) on the current device. Again, there are many gamut mapping algorithms, and a full discussion is outside the scope of this thesis. We use the approach of desaturating a color until it is within gamut throughout this thesis.

Chapter 3

Related Work

3.1 Analytical BRDF Models

Many analytic BRDF models have been created. Each model is appropriate for a specific class of materials. These models range from simple phenomenological models[18] to more complex physically based models[19]. The most common spectral analytical BRDF model is the dichromatic model[20], in which the BRDF is written as a linear combination of a diffuse and specular component: $f_r(\omega_i, \omega_o, \lambda) = g(\lambda) + h(\omega_i, \omega_o)$. However, this models separation of angular and spectral effects is too simple to capture many types of materials. For example, the Fresnel equation predicts the color of a specular highlight should change with the viewing angle. This can not be described by the dichromatic model. Furthermore, complex iridescent effects, such as those found in pearlescent pigments, are not describable by the dichromatic model.

3.2 BRDF Measurement Systems

A device for measuring a BRDF is called a *gonioreflectometer*. Many different designs for gonioreflectometers exist, each with their own set of strengths and weaknesses. Recall that the anisotropic spectral BRDF is a function of 5 angular variables and one wavelength variable $f_r(\theta_i, \phi_i, \theta_o, \phi_o, \lambda)$. Early gonioreflectometers were typically point based. Point based gonioreflectometers measure the BRDF at discrete angular positions sequentially. However, by using a spectroradiometer, the wavelength dimension(λ) can be captured in parallel. However, these devices can be very time consuming to use since a sampling device must be moved to many different points in a 5D domain sequentially. Contrastingly, image based systems use a camera to take many angular measurements of a BRDF simultaneously. However, image based gonioreflectometers have typically had low spectral resolution due to the lack of availability of multispectral cameras.

Early work on general purpose multispectral BRDF capture was conducted at the National Institutes of Standards and Technology(NIST)[21]. Their STARR system is capable of capturing full anisotropic multispectral BRDFs that ranging from UV to NIR in the spectral domain. However, each of the 4 angular dimensions is sampled sequentially, and thus dense angular sampling of a BRDF is infeasible due to long acquisition times. The first spectral BRDFs measured specifically for computer graphics are from the work of Li et al.[5]. The authors report being able to measure a full multispectral BRDF in 10 hours, with roughly 103 angular samples. In contrast to our approach, the authors capture the wavelength dimension in parallel since they use a spectroradiometer. We capture the wavelength dimension sequentially, but capture 2 angular dimensions in parallel.

Until Ward’s seminal paper[22], BRDF capture systems in computer graphics were slow in that the angular dimension was captured sequentially. In Ward’s work, the imaging gonireflectometer was introduced. Ward’s device, using a mirrored dome and a fisheye lens, allows the user to capture 2D angular slices of an anisotropic BRDF. However, Ward’s device has trouble with both grazing measurements and highly specular materials. Later, Steve Marschner’s paper on fast BRDF acquisition[3] made image based capture more practical. This setup is much simpler to use and more practical than Ward’s. The insight in this work is that curved material samples(as opposed to planar samples) allow camera based measurement devices to take one BRDF measurement at each pixel. This allows for faster acquisition. Later, Matusik et al. demonstrated the usefulness this type of design by capturing 100 RGB BRDFs using a design similar to that described by Marschner et al.[4].

More recently, Hullin and his collaborators built an image-based system capable of capturing isotropic bispectral BRDFs[15]. Bispectral BRDFs allow for fluorescence. Hullin and his collaborators also used an image based design similar to that of Marschner[3] and our own.

Chapter 4

Spectral Camera Calibration

4.1 Camera Calibration Goals

Camera calibration is the general process of determining some qualities about a camera, and then using these qualities to turn pixel values into an image into meaningful measurements about the real world. The definition of meaningful varies across applications. In consumer photography, the user simply wishes to produce an image that has pixel values, which, when displayed on a monitor, evoke a similar sensation to viewing the original scene. Contrastingly, in scientific applications, the user wishes to calibrate the imaging system such that pixel values in each spectral channel are proportional to spectral radiance at a particular point on an object.

Camera calibration is an important aspect of any image based BRDF measurement system. In our case, since we are using custom optical components to create a multispectral camera, the calibration task is even more challenging as compared to calibrating a traditional trichromatic camera. Our final goal is to be able to extract, at any given pixel, the underlying spectral reflectance of object at that location.

Camera calibration techniques fall into two categories: geometric calibration and radiometric calibration. Geometric calibration allows us to determine how pixels in an image correspond to points in the real world. Typically these algorithms solve for camera parameters that allow us to place the camera in a world coordinate system, and then trace rays through individual pixels. Radiometric camera calibration algorithms turn digital pixel values into meaningful radiometric quantities.

Geometric calibration is concerned with determining the geometric properties of the cameras optical system, and determining the camera's location in some world coordinate system. These two ingredients allow us to find a correspondence between pixels on the image plane and their location in the physical world.

Geometric camera calibration consists of determining the camera’s intrinsic and extrinsic parameters[23]. Intrinsic parameters don’t vary as we move the camera, and include the focal length(or equivalently the field of view), the pixel width and height, and the optical center on the image plane. The extrinsic parameters of a camera allow us to place the camera in a world coordinate frame. Typically, extrinsic parameters are given by a translation and a rotation that place a camera at the origin looking down the positive z axis. Our geometric calibration techniques are not new, and we use standard algorithms and tools[24].

Radiometric calibration is concerned with converting pixel values on an image into a meaningful radiometric unit, such as spectral radiance. This process is challenging due to spatial non-uniformities across the image plane(vignetting), dynamic range limitations of the sensor, and unknown quantum efficiency of the detector and any optical filters. In this thesis we focus primarily on radiometric calibration, since that is where we have made a novel contribution. Therefore, throughout the rest of the thesis when the term camera calibration is used, we refer to radiometric camera calibration unless explicitly noted otherwise.

Careful spectroradiometric calibration is essential if we are to extract meaningful measurements from our image based capture system. We wish to recover, at each pixel, a spectral reflectance that corresponds to the spectral reflectance of the material being imaged as closely as possible.

4.1.1 Mathematical Model

Notation	Meaning
$p_j(x, y)$	Pixel value at location (x, y) , tuning wavelength λ_j .
F	Camera transfer function.
t	Exposure time.
a	Pixel area.
$S(x, y)$	Spatial response of sensor.
$L(\lambda)$	Light spectrum.
$f(\lambda)$	Spectral Reflectance of material.
$O(\lambda)$	Spectral transmission of optics.
$V_j(\lambda)$	Spectral transmission of LCTF when tuned to λ_j .
$C(\lambda)$	Quantum efficiency of CCD.
$H_j(x, y)$	HDR value for LCTF tuned to λ_j at pixel (x, y) .
Λ	The visible spectrum. $\Lambda = [400, 720]\text{nm}$.

Table 4.1: Table of mathematical notation for calibration

In order to develop effective calibration techniques, a mathematical model of image formation must be developed. Our model specifies the value for a pixel $p_j(x, y)$, where (x, y) denotes the pixel’s position on the sensor, and j denotes the j th tuning wavelength of the Varispec filter. Therefore, $p_j(x, y)$ is the pixel value at spatial location (x, y) when the Varispec filter is tuned to it’s j th tuning wavelength, λ_j .

$$p_j(x, y) = F \left(t a S(x, y) \int_{\Lambda} L(\lambda) f(\lambda) O(\lambda) V_j(\lambda) C(\lambda) d\lambda \right) \quad (4.1)$$

In this equation, $F : \text{irradiance} \rightarrow [0, 2^{12})$, is the function that maps irradiance at a pixel to a 12 bit camera pixel value. Because we use a monochrome camera, there is only 1 function responsible for converting from irradiance to a 12 bit value. Therefore, F is independent of wavelength and spatial location on the sensor. The variable t represents exposure time, whereas a is the area of a pixel.

The function S models spatial non-uniformity across the sensor, primarily caused by vignetting. We verified that this vignetting is not wavelength dependent. $L(\lambda)$ is the spectrum of our light source.

The transmission of the Varispec when tuned to λ_j is given by $V_j(\lambda)$, and the transmission of the lens and other optical components is $O(\lambda)$. The function $C(\lambda)$ is the quantum efficiency of our CCD. The quantity of interest, the materials spectral reflectance, is $f(\lambda)$. Note that the above equation ignores pixel crosstalk, which is a common and accepted simplification in multispectral imaging research[25].

We measure relative BRDFs, which, for graphics applications, are typically sufficient. There exists a single global scale factor that could be solved for to relate our BRDF data to real-world units. This scale factor could be determined by a method proposed by Hullin[15]. As a result not needing exact units, we can simplify equation 4.1 by removing a (pixel size) since it is constant for each measurement.

We are limited by the 12 bit dynamic range of our camera. The dynamic range between dark areas of a BRDF and a bright specular highlight is large. Regardless of what exposure time we choose, some areas will be saturated and others will be too dark. To remedy this issue we create a single high dynamic range image(HDR) for each light position/wavelength pair. We take multiple exposures, ranging from $10\mu\text{s}$ to 10 s. Using these multiple exposures, we then create an HDR image. Our camera has a highly linear response, and therefore we simply fit a line to our exposure time vs pixel value measurements, and take this slope as proportional to incident irradiance. We ignore pixel values in the toe and the shoulder of our cameras response curve, since in this region the response can be non-linear. We validated the linearity of our camera, we compute the Pearson r coefficient for a set of input images and always found that it was always greater than .99 over the region of interest. Note that linear response is common in scientific grade cameras, and this is the same approach taken by Matusik[6]. If our camera's response curve was non-linear, we could use methods such as those proposed by Debevec[26] to invert F .

Using the above simplifications, and grouping the spectral response of our system into $R_j(\lambda) = V_j(\lambda)O(\lambda)C(\lambda)$ we are left with the following equation:

$$H_j(x, y) \propto S(x, y) \int_{\Lambda} L(\lambda) f(\lambda) R_j(\lambda) d\lambda \quad (4.2)$$

Here $H_j(x, y)$ is the HDR pixel value at location (x, y) for spectral band λ_j . Our final goal is to obtain an accurate estimate of $f(\lambda)$. We divide this process into 3 distinct stages. The **spatial calibration** stage is concerned with measuring $S(x, y)$. **Spectral calibration** is the process of determining $R_j(\lambda)$. Spatial and spectral calibration only need to be done once for a given camera. Finally, the process of **spectral recovery** corresponds to solving for $f(\lambda)$ once spatial and spectral calibration have been performed. Spectral recovery is performed per-pixel.

4.1.2 Spatial Calibration

We use an integrating sphere to produce a uniform field of light. We then image this field of light to determine $S(x, y)$. The function $S(x, y)$ returns values in $[0, 1]$ where 1 indicates all light passes through, and 0 indicates that no light passes through. We do this by imaging a flat field of illumination using an integrating sphere. We do this for each tuning wavelength λ_j . A single image is produced at each wavelength, with whatever exposure time allows us to avoid sensor saturation. Our integrating sphere setup is shown in figure 4.1.

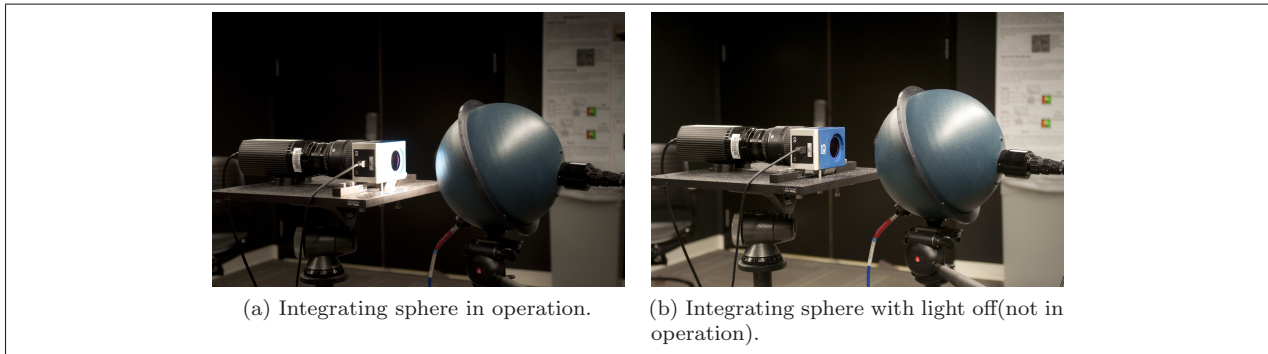


Figure 4.1: Calibrating the camera using an integrating sphere. This calibration accounts for vignetting and other spatial (but not spectral) non-uniformities across the sensor. The integrating sphere allows us to produce a uniform field of illumination across the camera sensor.

Next, for each image, we divide each pixel by the maximum pixel value taken from the center of the image. Now, each image has values in $[0, 1]$. Finally, we average the results from each tuning wavelength together, and manually verify that the vignetting pattern is not wavelength dependent.

One concern is that there will be a low signal to noise ratio in the corner of our images due to vignetting. While this may be an issue for general multispectral imaging, in our case it is not. This is because, for BRDF measurement, the spherical targets we measure are all near the center of the image.

4.1.3 Spectral Calibration

The goal of spectral calibration is to find a model for $R_j(\lambda)$. We have tested several models, including a **delta model** and a **factored model**. The delta model idealizes each $R_j(\lambda)$ as a delta function centered

at λ_j . This model is conceptually simple, and makes the spectral reconstruction phase very fast. As a second approach, we propose a **factored model**. This approach leverages that LCTF transmission data is provided by the manufacturer, and that CCD quantum efficiency curves are smooth.

For spectral calibration, we take M HDR images of objects of known reflectance. In each image we know have measured the illuminant's spectrum. In our case, $M = 5$ and we use 5 NIST traceable diffuse reflectance standards from Labsphere. Our targets are red, green, blue, yellow, and white(Spectralon). We denote these known spectral reflectances as $f'_1(\lambda) \dots f'_M(\lambda)$. Our calibration targets, as well as their spectral reflectance curves at normal incidence, are shown in figure 4.2.

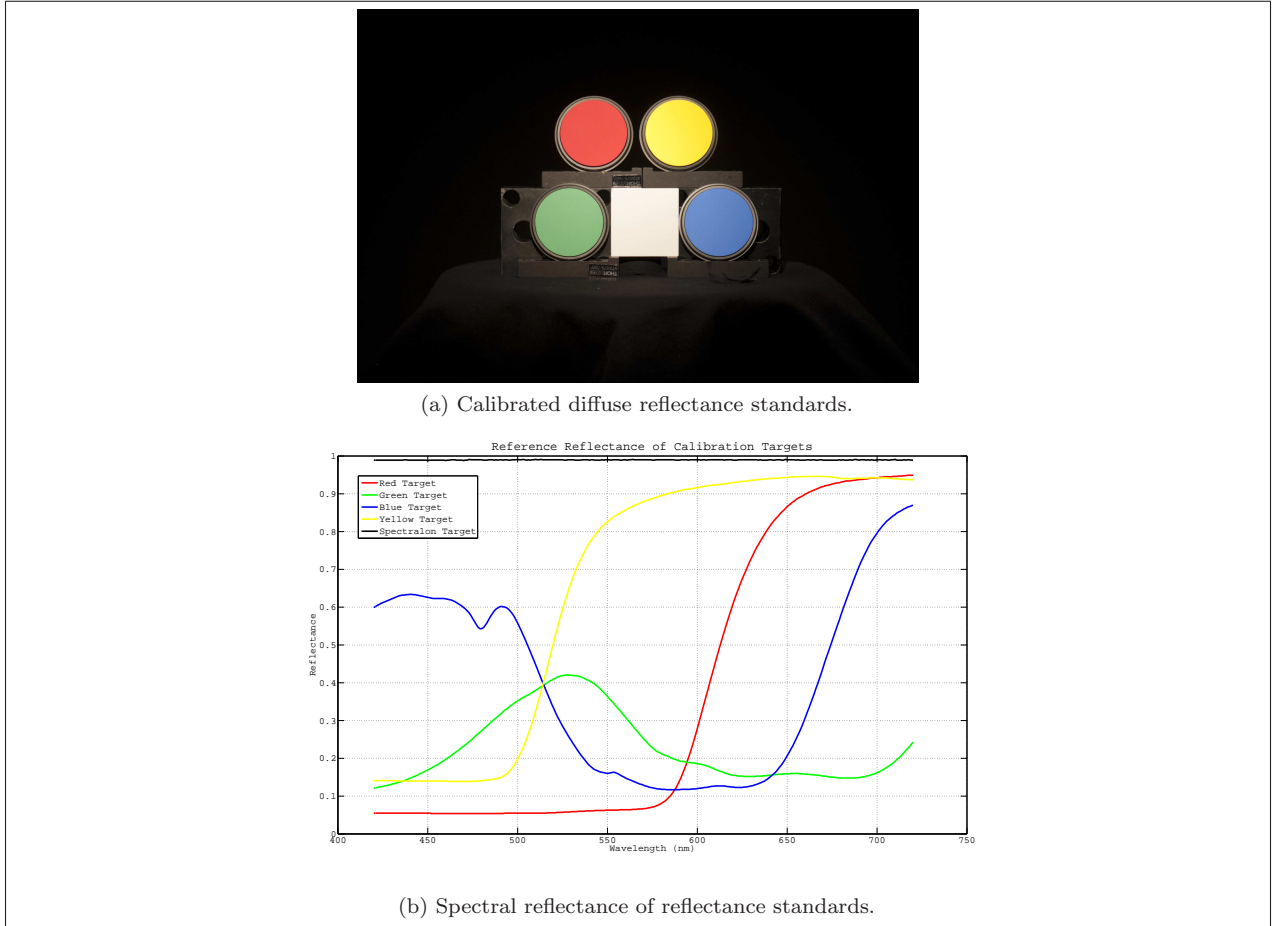


Figure 4.2: Spectral calibration targets. Reflectance is plotted at normal incidence($\theta_i = \theta_o = 0$), but materials are close to ideal Lambertian reflectors.

Since each reflectance target takes up a relatively small area in the center of the image plane, we temporarily ignore spatial calibration. Therefore, our image formation equation reduces to the following.

$$H_{ij} = \int_{\Lambda} L(\lambda) f'_i(\lambda) R_j(\lambda) d\lambda \quad (4.3)$$

Here i indexes over our different images/targets of known reflectance and j indexes over tuning wavelengths λ_j .

Delta Model

Our delta model considers the response of each tuning wavelength as an idealized impulse. Formally, we model $\tilde{R}_j(\lambda)$, our estimate of the spectral response, as follows:

$$R_j(\lambda) \approx \tilde{R}_j(\lambda) = w_j \delta(\lambda - \lambda_j) \quad (4.4)$$

When we insert equation 4.4 into equation 4.3, we see that the integral falls away and we are left with:

$$H_{ij} = L(\lambda_j) f_i(\lambda_j) w_j \quad (4.5)$$

Therefore, for each single unknown w_j we setup an overdetermined system of M equations and solve to find the parameter w_j , the height of the delta function for tuning wavelength λ_j . A visualization of the camera response modelled as delta functions is shown in figure 4.3.

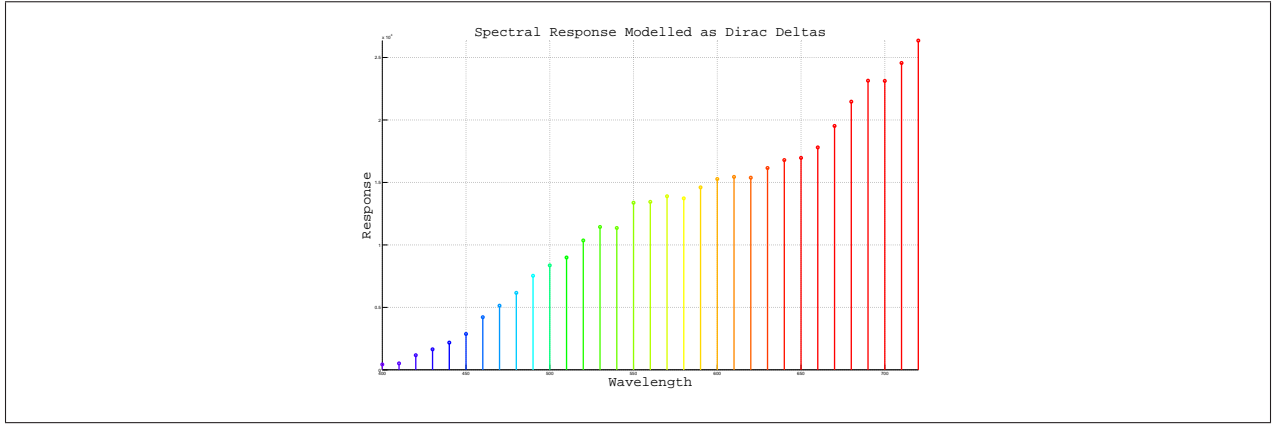


Figure 4.3: $R_j(\lambda)$, the Spectral transmittance of Varispec multiplied by Quantum Efficiency modelled as Dirac delta functions.

Factored Model

The factored model takes advantage of the facts that CCD quantum efficiency ($C(\lambda)$) and optical transmission ($O(\lambda)$) curves are highly smooth, and that the manufacturer of the our LCTF provides tabulated data for spectral transmission for certain tuning wavelengths. We then choose to model $R_j(\lambda)$ as the product of the manufacturer's LCTF transmission data and a constant. The manufacturer's tabulated data is given by $T_j(\lambda)$. Our model then becomes:

$$R_j(\lambda) \approx \tilde{R}_j(\lambda) = c_j T_j(\lambda) \quad (4.6)$$

We solve for c_i by pulling it out of the integral and using numerical quadrature to find $\int_{\Lambda} T_j(\lambda) L(\lambda) f(\lambda) d\lambda$. A visualization of the camera response modelled via the factored model is shown in figure 4.4.

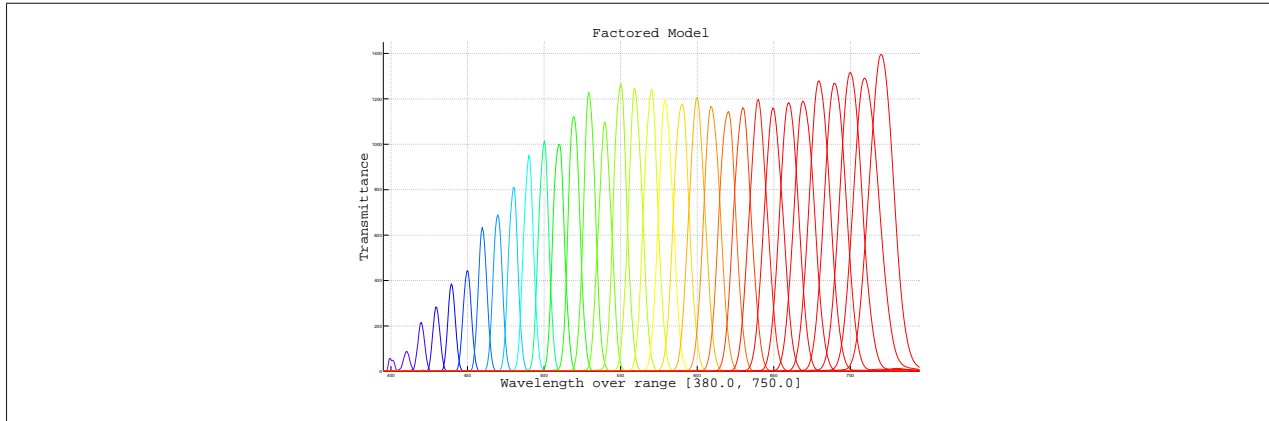


Figure 4.4: Plot of $c_j T_j(\lambda) \forall j$. Note the large width of $T_j(\lambda)$ near $j = 28$ and above.

4.1.4 Spectral Reconstruction

The process of spectral reconstruction consists of estimating $f(\lambda)$, the spectral reflectance, given known calibration parameters. As shown in figure 4.5, the reflectances of real world objects are vary rapidly, and thus this reconstruction is a challenging process.

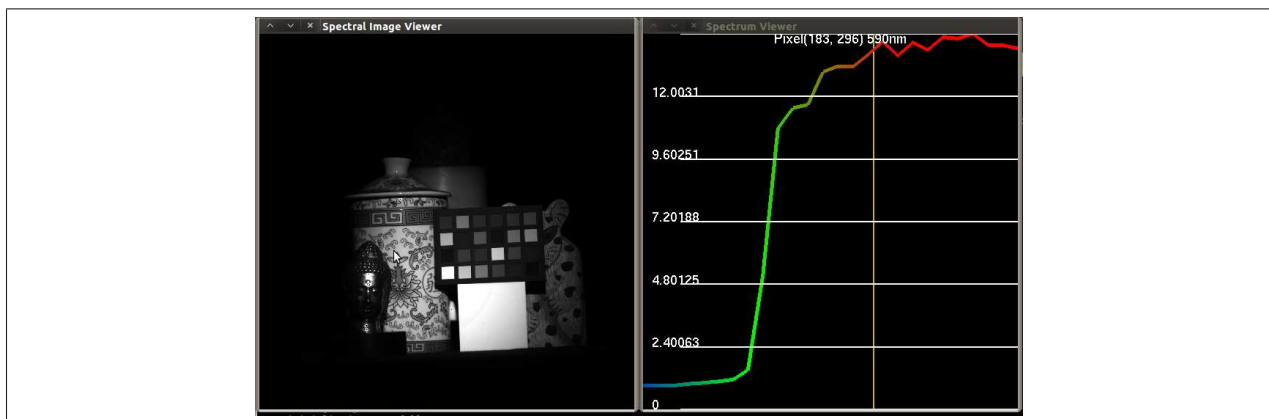


Figure 4.5: A screenshot of our spectral image viewer showing the spectral reflectance of a yellow vase. This application allows a user to move the mouse over the pixels on the right hand side of the image. The spectral reflectance underneath the mouse is shown on the right hand side in real-time. The black and white image on the left is a visualization of the intensity at the wavelength represented on the right by a yellow vertical line.

4.1.5 Nyström Reconstruction

For any spectral calibration model, the Nyström method can be used to reconstruct the spectral reflectance. In the Nyström method a numerical quadrature rule is discretized into a system of linear equations. In our case, this discretization yields the following linear system.

$$\begin{bmatrix} R_1(\lambda_1) & \dots & R_1(\lambda_N) \\ \vdots & & \vdots \\ R_N(\lambda_1) & \dots & R_N(\lambda_N) \end{bmatrix} \begin{bmatrix} L(\lambda_1) & & \\ & \ddots & \\ & & L(\lambda_N) \end{bmatrix} \begin{bmatrix} f(\lambda_1) \\ \vdots \\ f(\lambda_N) \end{bmatrix} = \begin{bmatrix} H_1 \\ \vdots \\ H_N \end{bmatrix} \quad (4.7)$$

For Nyström reconstruction, we invert the matrix \mathbf{RL} , and multiply this by each spectrum in order to reconstruct the reflectance spectrum. We exploit the fact that $(\mathbf{RL})^{-1} = \mathbf{L}^{-1}\mathbf{R}^{-1}$. Note that for the delta model, the matrix \mathbf{RL} is a diagonal matrix, so this inversion process is trivial. For the factored model, this matrix is not a diagonal matrix.

4.1.6 Multispectral Imaging Results and Comparison

Evaluation of calibration and spectral reconstruction methods is challenging. The primary issue is that limited ground truth data is available on which to test spectral reconstruction techniques. Secondly, the ground truth data that is available is of very smooth spectral reflectance, which most techniques perform well on. Finally, the question of what spectral comparison metric to use is highly application dependent [27].

Weighted comparison metrics require knowledge of the lighting in which the spectra will be viewed, and the spectral response of the viewer. Since we do not know all situations in which our data will be used, we can not choose a weighted metric. As a result, we choose RMS, which is commonly used in graphics [28]. Our metric to compare two spectra S_1 and S_2 is given below.

$$\Delta_{\text{RMS}}(S_1, S_2) = \sqrt{\frac{\sum_{i=1}^N (S_1(\lambda_i) - S_2(\lambda_i))^2}{N}} \quad (4.8)$$

We fit both of our models from data on the blue and white(Spectralon) targets. We then tested them on the red, yellow, green and blue targets. Despite its simplicity, the delta model was superior on all test data. Results are shown in figures 4.6, 4.7, 4.8, and 4.9.

Perhaps counterintuitively, the delta model performed better on all input data. This is surprising, since this model does not account for the width of the LCTF transmission curves. One possibility is that the data provided by the manufacturer is erroneous, or that our light data measured by the spectrometer is incorrect. A final possibility is that the CCD QE is not as smooth as expected.

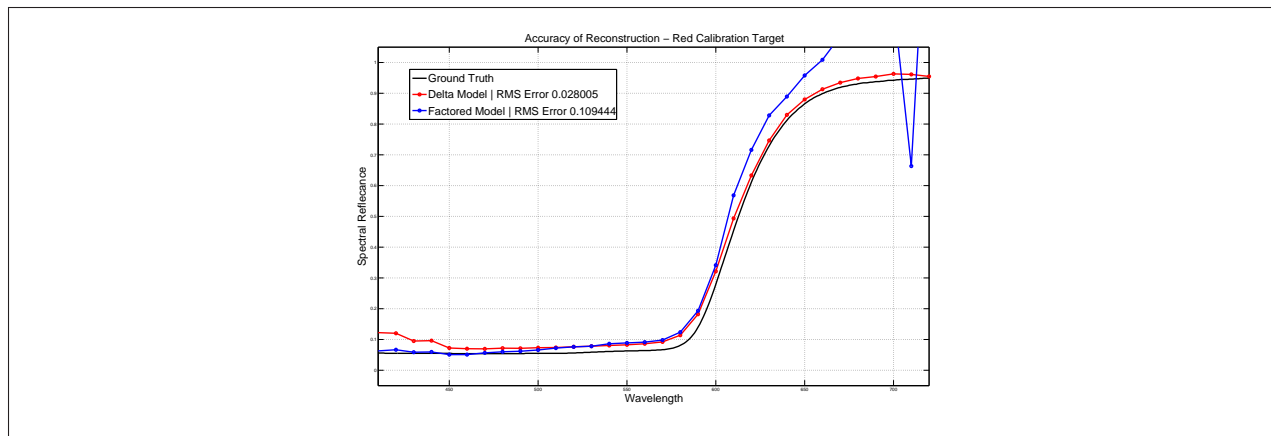


Figure 4.6: Comparison of spectral calibration techniques using the red target. Results for both the factored model and the delta model are shown.

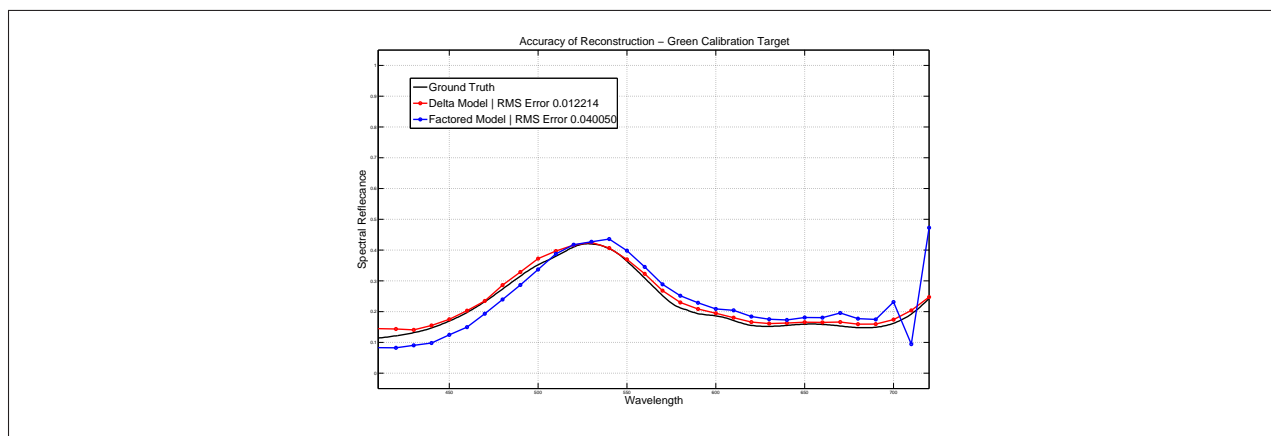


Figure 4.7: Comparison of spectral calibration techniques using the green target. Results for both the factored model and the delta model are shown.

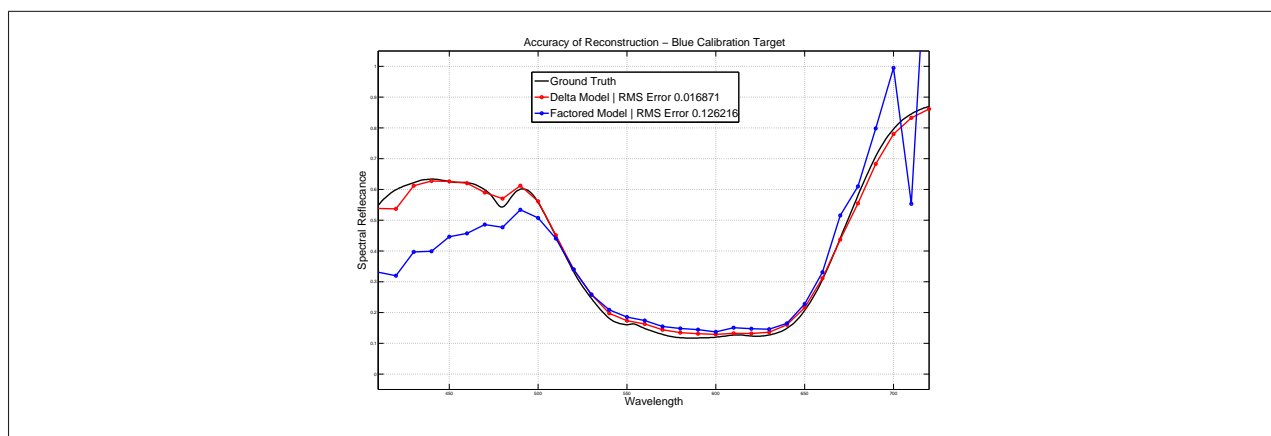


Figure 4.8: Comparison of spectral calibration techniques using the blue target. Results for both the factored model and the delta model are shown.

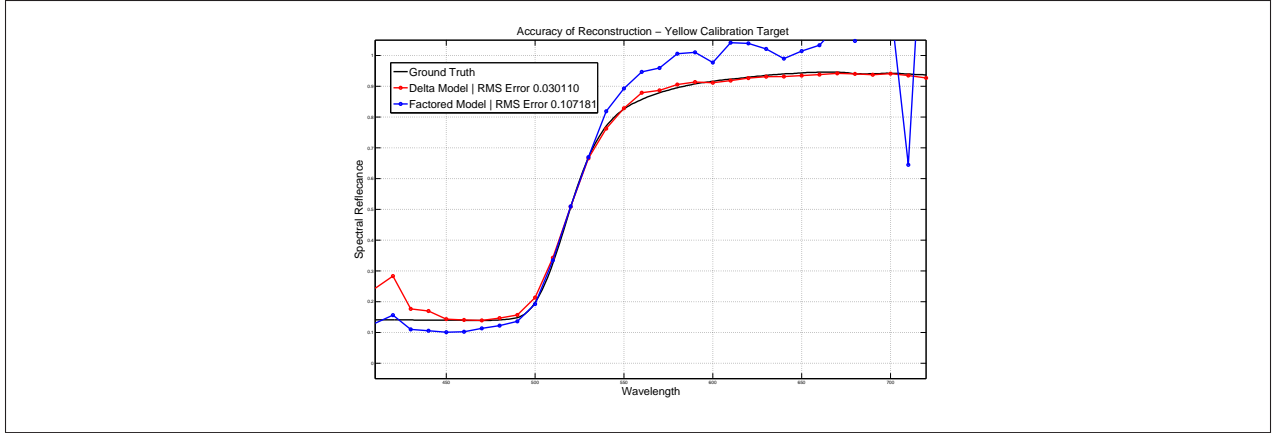


Figure 4.9: Comparison of spectral calibration techniques using the yellow target. Results for both the factored model and the delta model are shown.

4.1.7 Multispectral Relighting Results

One important application of multispectral imaging is relighting [28]. Relighting is the process of taking an image of a scene under one illuminant, and then synthesizing artificial images of the same scene under a different illuminant. Multispectral images are of paramount importance for accurate relighting. Consider the situation in which an image is taken under an illuminant with known spectrum $L(\lambda)$, and one wishes to relight the scene with a different illuminant $L'(\lambda)$. The RGB values for the relit image will be R' , G' , B' . For multispectral data, relighting is given by the following equation: are computed as follows:

$$\begin{aligned} R' &= \int_{\Lambda} \bar{r}(\lambda) \frac{L(\lambda)f(\lambda)}{L(\lambda)} L'(\lambda) d\lambda \\ G' &= \int_{\Lambda} \bar{g}(\lambda) \frac{L(\lambda)f(\lambda)}{L(\lambda)} L'(\lambda) d\lambda \\ B' &= \int_{\Lambda} \bar{b}(\lambda) \frac{L(\lambda)f(\lambda)}{L(\lambda)} L'(\lambda) d\lambda \end{aligned} \quad (4.9)$$

Note that in these equations, $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, and $\bar{b}(\lambda)$ are the red, green, and blue spectral matching curves, respectively. With multispectral relighting, we get an accurate result since we can divide out the contribution of the original light, and factor in the contribution of the new light.

In a typical, tristimulus relighting pipeline, an image is acquired with an RGB camera. The acquisition process corresponds to integrating with the spectral matching curves. Therefore, relighting can only operate on RGB values, and not on spectral data. Relighting in RGB is given by the following equation, where

R, G, B specify the original color values before relighting:

$$\begin{aligned} R' &= \frac{R}{\int_{\Lambda} L(\lambda) \bar{r}(\lambda) d\lambda} \int_{\Lambda} \bar{r}(\lambda) L'(\lambda) d\lambda \\ G' &= \frac{G}{\int_{\Lambda} L(\lambda) \bar{g}(\lambda) d\lambda} \int_{\Lambda} \bar{g}(\lambda) L'(\lambda) d\lambda \\ B' &= \frac{B}{\int_{\Lambda} L(\lambda) \bar{b}(\lambda) d\lambda} \int_{\Lambda} \bar{b}(\lambda) L'(\lambda) d\lambda \end{aligned} \quad (4.10)$$

The benefits of multispectral relighting are demonstrated in figure 4.10.

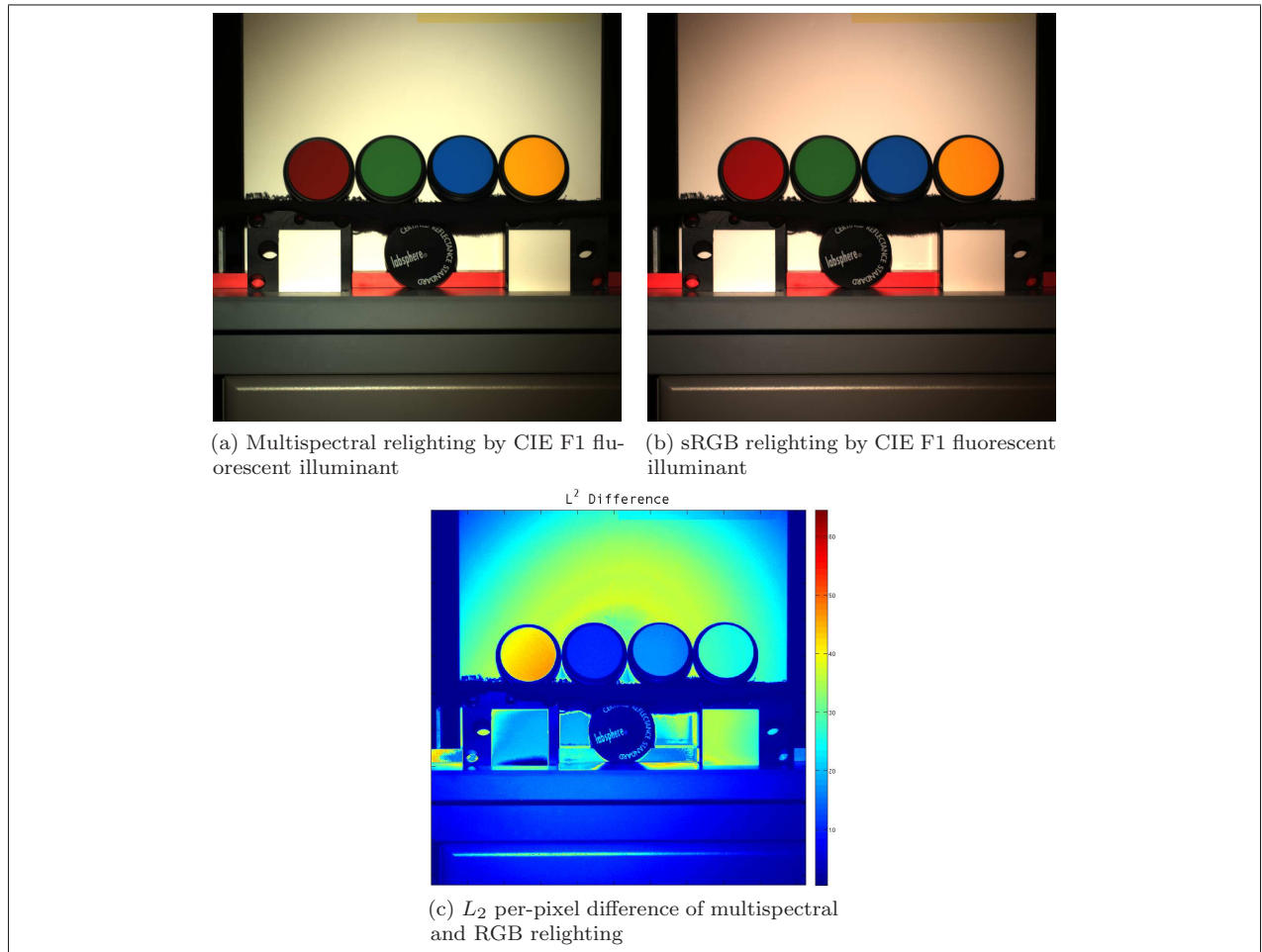


Figure 4.10: Comparison of multispectral and RGB relighting results.

Chapter 5

BRDF Measurement

5.1 Measurement Algorithm Overview

To measure multispectral isotropic BRDFs, we have built an image based gonireflectometer similar in principal to the one originally designed by Marschner et. al[3]. Our image based gonireflectometer is also similar to that of Matusik et al.[4]. A diagram of our system is shown in figure 5.1. We move the light source in a 165 degree arc around the sample in 1 degree increments. We are unable to measure a full 180 degree arc due to the light source obstructing the camera, and due to the light source shining directly into the lens. We could measure a larger range of angles if we had used a half-silvered mirror, but as this cuts light transmission in half and complicates the system design we opted against it. By capturing only an 165 degree arc, as opposed to 180, we limit our ability to measure retroreflection and grazing configurations of the BRDF. Both Matusik[4] and Marschner[3] reported the same issue. At each light position, we create an HDR image at several wavelength bands. Currently we take 31 images, ranging from 420nm to 720nm in 10nm increments. We don't capture below 420nm due to the low transmission our LCTF in the near-UV range.

At each light position, we create an HDR image for each spectral band. We take a series of up to 20 exposures, ranging from 100 microseconds to 8 seconds. In most spectral bands, 8 second exposures are not necessary. Before measuring a BRDF, the user supplies a series of constraints stating the maximum exposure time necessary at each wavelength band. We have found that long exposures are typically only necessary below 480nm. To compute each HDR image we simply fit a line to our exposure time vs. pixel value measurements. We use the slope of this line as our radiance estimate. We discard all measurements with pixel values below 15 and above 240. We are able to use the slope of this line as a radiance estimate

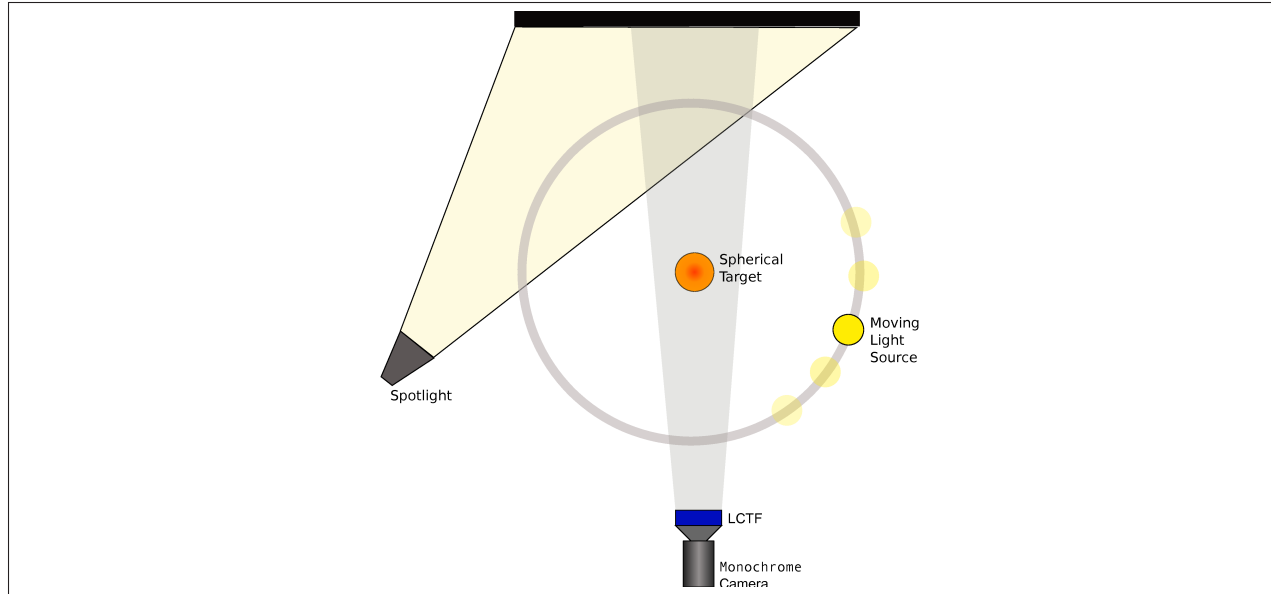


Figure 5.1: Diagram of image based BRDF measurement system using spherical samples.

due to the linearity of our camera's transfer function (CTF) over the range $[15, 240]$. If our CTF was not linear, we could use methods like that proposed by Debevec et. al[26]. to invert the CTF.

Our system is different than that of previous work in a few ways. First, before taking BRDF measurements, we turn off the moving light source and turn on the spotlight. The spotlight illuminates a board behind the spherical target, and creates an image in which outline of the sphere is dark, and the background is white. This allows us to separate which pixels are on the sphere versus which are part of the background. We also use data from this image to derive per-pixel normal vectors, light vectors, and view vectors. This aspect of our system is described further in section 5.2. Our setup is easier to use and cheaper than that of Matusik et al[6], who uses a touch-probe to determine the exact distance from the camera to the sphere. Our system requires no physical measurements of this kind.

5.1.1 Components of BRDF Measurement System

Our system uses a QImaging Retiga 4000R cooled CCD monochrome camera with a Nikon lens to capture images. We have removed the IR filter from this camera since it limited our ability to capture above 680 nm. Noise from heat is limited by an IR cut filter on our light source that operates closer to 730 nm. To take multispectral images we use a Varispec VIS LCTF (10 nm bandwidth) made by Caliper Life Sciences. Our light source is a 150 W Tungsten Halogen stable source produced by Schott-Fostec (Schott Fostec DCR-III). Images of our physical setup are shown in figure 5.2.

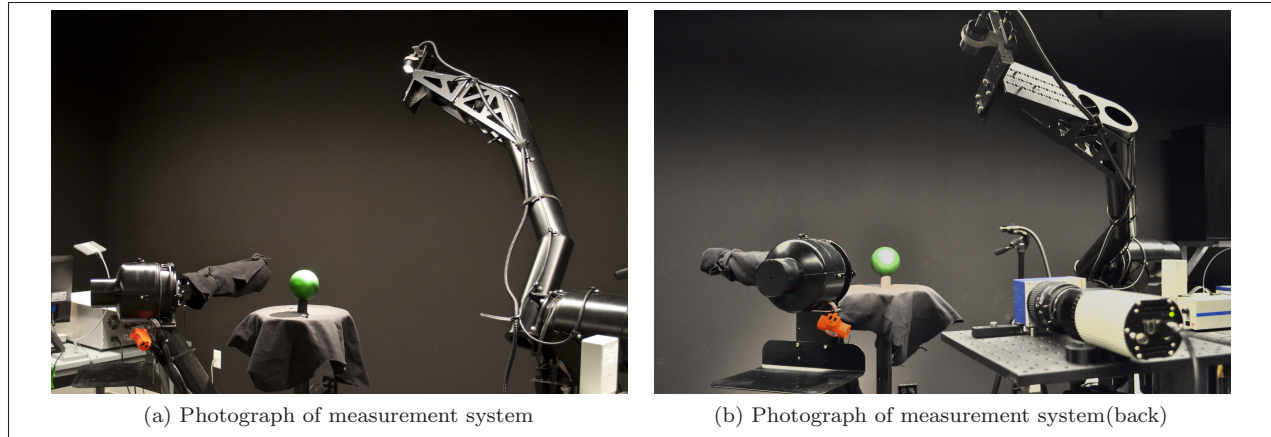


Figure 5.2: Photographs of BRDF measurement system.

We were initially concerned with polarization affecting our measurements. Light from our light source is unpolarized. However, BRDFs will sometimes linearly polarize light. This polarization typically occurs around the specular highlight. Our LCTF acts as a linear polarizer. Light that is linearly polarized with the correct orientation will have twice the transmission of unpolarized light after passing through a linear polarizer. If this was to occur, this would cause us to measure the highlight incorrectly. One way to avoid this, which was done by Hullin et al.[15], is to place an optical depolarizer in front of the LCTF. We tried this with two depolarizers, but did not get good results. First, these depolarizers cut down on transmission of light, and thus increase required integration times, which amplifies noise. Furthermore, only small optical depolarizers are readily available, which then causes severe vignetting and cuts down on the amount of pixels available as measurements. We had also tried a custom made large aperture depolarizer, but the AR coating on this component was not effective at reducing internal reflections. These internal reflections corrupted our data and made the depolarizer unusable.

5.2 Converting HDR values to BRDF Measurements

Once we have captured our data, made HDR images from LDR input, and performed spectral calibration, we are left with a series of multispectral HDR images for various light positions. Pixels on the sphere itself are BRDF measurements, but pixels not on the sphere (or in shadow) must be discarded. To initially determine which pixels are located on the sphere we use the silhouette image taken during the capture process. This image is taken with a backlight on, and the moving light source off. This results in an image where the sphere itself is dark, but the background is white. In order to detect the boundaries of the target sphere,

we need an algorithm for detecting a circle in the image¹. We minimize the following expression using the Nelder-Mead simplex algorithm[29] to determine the center pixel (C_x, C_y) as well as the radius in pixels, r .

$$f(C_x, C_y, r) = - \sum_{i=1}^N \left(\left| \nabla I \left(C_x + r \sin\left(\frac{2i\pi}{N}\right), C_y + r \cos\left(\frac{2i\pi}{N}\right) \right) \right| \right) \quad (5.1)$$

In this minimization f is the objective function to minimize, N is the number of sample points (we use $N = 32$), and ∇I is the gradient of the monochromatic silhouette image. This minimization is attempting to place a circle on the image in a location where the gradient magnitude is large around the edge of the circle. A step of this minimization is visualized in figure 5.3, where the gradient magnitude is visualized. The green circles are the N points at which gradient magnitude is evaluated, and the red circle is a visualization of the current iteration's best guess.

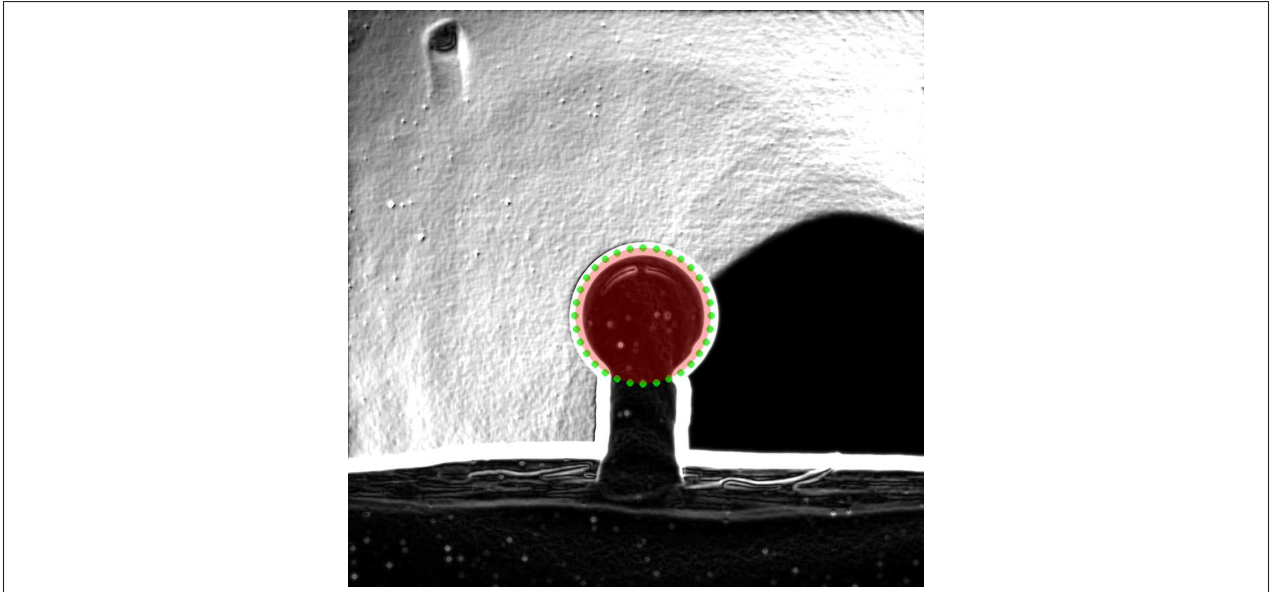


Figure 5.3: Visualization of non-linear minimization used for circle detection. The green points represent the N locations at which we sum the gradient magnitude. The gradient magnitude is visualized in black and white in the background. Note that the 8-bit imagery limits our ability to see subtle variations in gradient magnitude that are present in the underlying data.

5.2.1 Interpreting Pixel Values as BRDF Measurements

After performing circle detection, we know which pixels lie on the sphere. However, in order to interpret each pixel value as a BRDF measurement we must know the pixel's normal vector(\hat{n}), light vector($\hat{\omega}_i$), and view vector($\hat{\omega}_o$). Once these vectors are known, we have enough data to interpret each HDR value as a BRDF

¹In general, a sphere being imaged under perspective projection does not produce a circle. However, as our camera is near orthographic and our sphere is near the center of projection, our images are highly circular.

measurement. We determine surface normals and view vectors via raycasting. We determine the light vector either by imaging a mirrored sphere.

5.2.2 View Vector Determination

We initially perform geometric camera calibration using the method of Zhang et. al[24]. We take images of a checkerboard under various configurations and determine the intrinsic properties of our camera. Once our camera intrinsics are known, we assume the camera COP lies at the origin of a world coordinate system looking down the positive z axis. Using raycasting with a perspective camera model, we determine per pixel view vectors for our camera.

5.2.3 Surface Normal Determination

The sphere we are imaging lies down the z axis, roughly near the z axis and likely slightly off center. Once we know the sphere radius and position we can use raycasting to generate surface normals.

We would like to solve for (S_x, S_y, S_z, r) , the center of the sphere, and its radius, in world coordinates. We supply an initial guess, and use non-linear optimization[29] to minimize a 4 dimensional objective function g . Our objective function to be minimized is of the following form:

$$g(S_x, S_y, S_z, r) = \sum_{x=1}^W \sum_{y=1}^H (I_{\text{reference}}(x, y) \otimes I_{\text{traced}}(x, y)) \quad (5.2)$$

where $I_{\text{reference}}$ and I_{traced} are binary images where each pixel indicates whether the sphere is visible. The image I_{traced} is computed based on raytracing using the sphere given by the current value of (S_x, S_y, S_z, r) . The image $I_{\text{reference}}$ is created using the results from minimizing equation 5.1. The operator \otimes denotes exclusive or(xor). Given a reasonable initial guess, this algorithm converges in a few hundred iterations and finds a sphere placement and size that, when raytraced, results in exactly the same set of pixels as those found in the silhouette detection step. Once we have the placement and radius of our sphere we raycast to determine per-pixel normal vectors.

5.2.4 Light Vector Determination

Determining the light vector($\hat{\omega}_i$) per pixel is more challenging than determining the view vector or the normals. We idealize our light source as an infinitely distance point source for light vector determination. With this model, the light vector is constant over the entire sphere. We take an initial set of images of a mirrored sphere using our moving light source. We manually set the exposure time such that the specular

highlight on the sphere is under 3 by 3 pixels in size. Next, in each image, we use template matching to compute the location of the specular highlight. Once we have a particular pixel $(x, y) \in \mathbb{Z} \times \mathbb{Z}$ at which the specular highlight is located, we use the following equation to find our light vector $\hat{\omega}_i$:

$$\begin{aligned}\hat{\omega}_{ix} &= (2\hat{n} \cdot \hat{\omega}_o)\hat{n}_x - \hat{\omega}_{ox} \\ \hat{\omega}_{iy} &= (2\hat{n} \cdot \hat{\omega}_o)\hat{n}_y - \hat{\omega}_{oy} \\ \hat{\omega}_{iz} &= (2\hat{n} \cdot \hat{\omega}_o)\hat{n}_z - \hat{\omega}_{oz}\end{aligned}\tag{5.3}$$

We then store our these values into a precomputed table of values and look them up when processing future BRDF scans.

5.2.5 Unstructured BRDF Data Results

At this point we have, at each pixel in our image sequence, a value proportional to spectral reflectance. Furthermore, we have per-pixel light($\hat{\omega}_i$), view($\hat{\omega}_o$), and normal(\hat{n}) vectors, we have enough information to create BRDF measurements. By simply looping over each pixel on the sphere in our image sequence, we can store BRDF measurements. However, these measurements are *unstructured* in that they are scattered throughout the 3D space $(\theta_i, \theta_o, \phi_{\text{diff}})$. However, they are **not** unstructured in the wavelength dimension, λ , since we have sampled at regular 10 nm intervals.

Using these unstructured data points, we can create goniometric plots to visualize the BRDF. One form of goniometric plots is an incidence plane plot. The *incidence plane* of the BRDF is the 2D slice of the BRDF's angular domain shown in figure 5.4. Incidence plane plots are useful as they allow us to visualize many interesting features of the BRDF(ideal specular reflection, grazing, and retro-reflection) simultaneously.

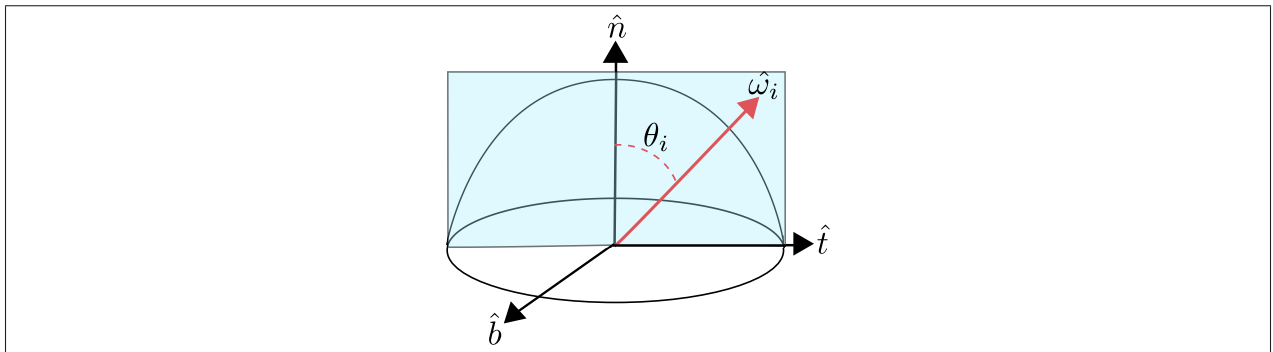


Figure 5.4: The incidence plane of the BRDF. In this diagram, the incidence plane is shown in light blue. The vectors \hat{n} , $\hat{\omega}_i$ and \hat{t} lie inside the incidence plane, while \hat{b} is normal to the incidence plane. In our incidence plane plots, we plot all BRDF samples such that $\hat{\omega}_o$ is near the incidence plane.

Each of our unstructured BRDF samples lies at a particular position. None will likely lie *exactly* in the incidence plane. Therefore, in our incidence plane plots we plot all BRDF measurements such that they are within some angular distance ϵ of the incidence plane. Each of our measured BRDFs has a different number of samples, owing to each sphere taking up different sizes in the images and due to pixels being rejected due to noise or bloom. Therefore, ϵ must be hand-tuned on many of the plots in order to get illustrative plots. Also, note that for most of our plots we plot the cube-root of the BRDF, as opposed to the BRDF itself. This is because the specular highlight of many materials is orders of magnitude larger than the diffuse component. This allows us to inspect both the diffuse and specular component simultaneously. This is the same practice used by Nagan et al.[30].

Below in figure 5.5 we see incidence plane plots for various angles θ_i for a glossy red Christmas ornament.

In figure 5.6 we show incidence plane plots for a highly specular red plastic material. Note sharp specular highlight.

In figure 5.7 we show incidence plane plots for blue racquetball. The BRDF is nearly Lambertian. There are large amounts of data missing near retro-reflection since this data came from an early version of our system with a different physical setup.

5.3 BRDF Reconstruction from Scattered Data

After the previous steps are complete, we now have a set of unstructured BRDF samples. Recall that the isotropic multispectral BRDF is a function of 3 angular variables and one wavelength variable: $f_r(\theta_i, \theta_o, \phi_{\text{diff}}, \lambda)$. After the previous steps have completed, we have N scattered BRDF measurements. Each measurement gives a full spectral sampling of the BRDF at one particular angular orientation $(\theta_i, \theta_o, \phi_{\text{diff}})$. This representation of the BRDF is not useful for either a rendering system or for efficient analysis. We would like a representation in which the BRDF is sampled onto a set of predetermined grid points. Such a representation allow constant time BRDF evaluation in a rendering system, and makes data analysis and visualization faster. Therefore, a procedure for resampling the BRDF onto a regular grid is required.

We use the non-uniform grid format that is used in the MERL BRDF database[6]. This format stores the BRDF using a halfangle-difference parameterization originally proposed by Rusinkiewicz[31]. This parameterization allows us to vary the sampling density near the specular highlight. This is impossible with the standard parameterization $(\theta_i, \theta_o, \phi_{\text{diff}})$. Using this parameterization, we sample the BRDF at $90 \times 90 \times 180$ positions, with more samples near the specular highlight. This allows us to reproduce more mirror-like surfaces.

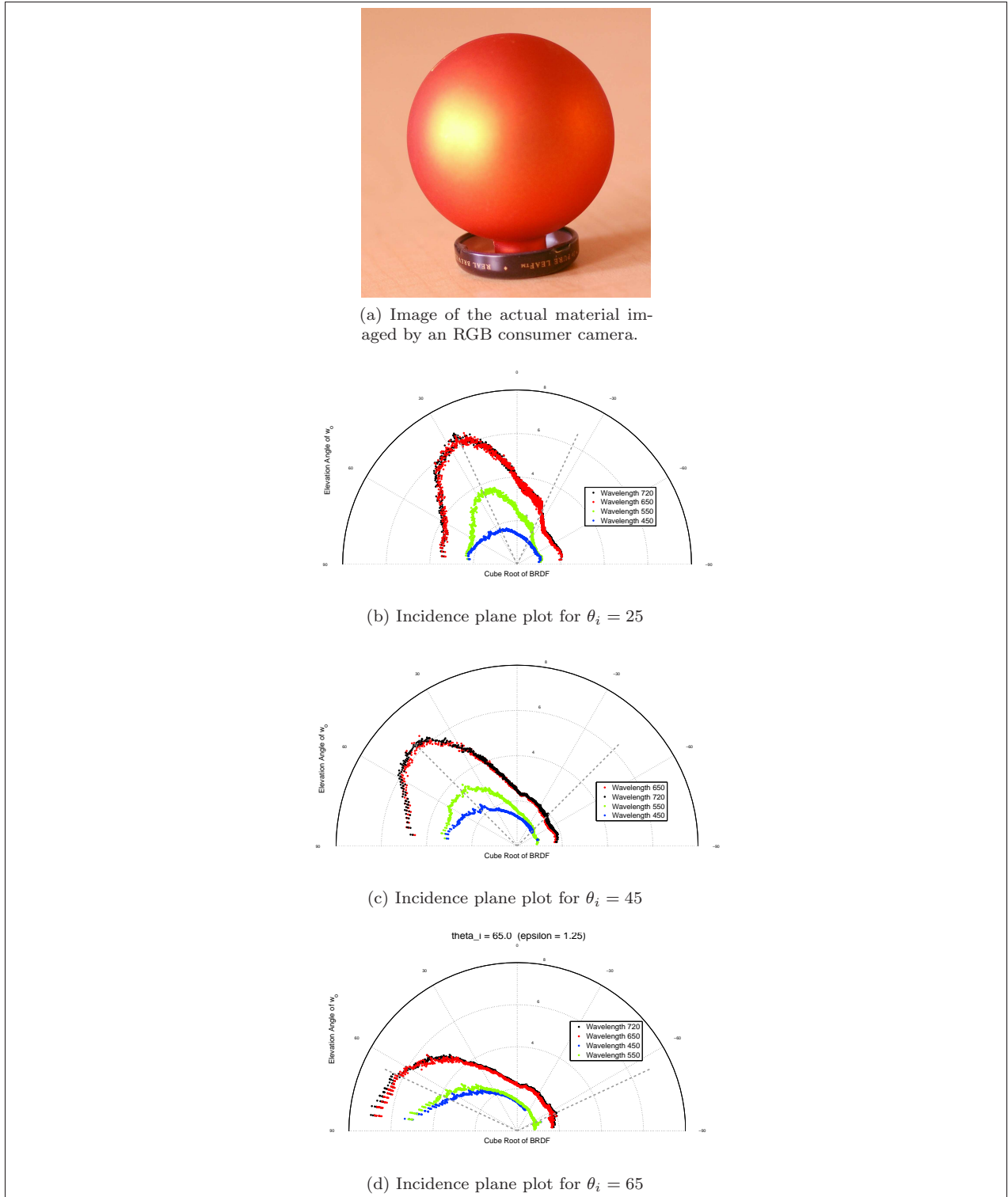
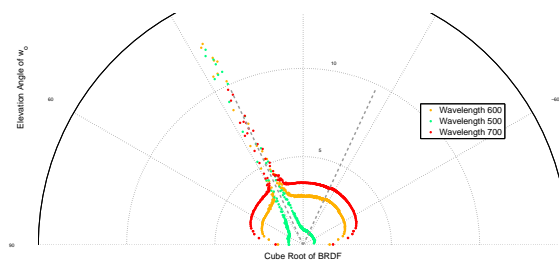


Figure 5.5: Incidence plane BRDF plots of a reddish-orange Christmas ornament. Note the glossy highlight.

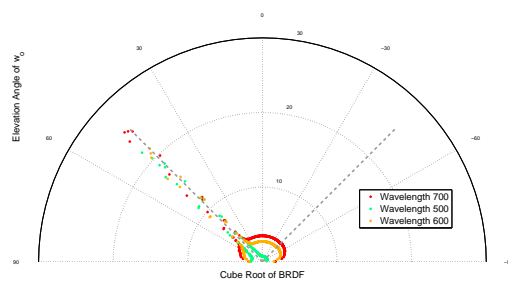
Formally, we have N samples of the BRDF, f_r , at coordinates $p_j = (\theta_{i,j}, \theta_{o,j}, \phi_{\text{diff},j})$ for $j = 1, 2, \dots, N$. At each angular position, we have a corresponding BRDF measurement b_j , where $b_j = f_r(\theta_{i,j}, \theta_{o,j}, \phi_{\text{diff},j})$.



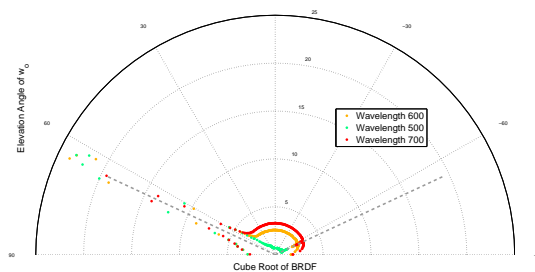
(a) Image of the actual material imaged by an RGB consumer camera.



(b) Incidence plane plot for $\theta_i = 25$



(c) Incidence plane plot for $\theta_i = 45$



(d) Incidence plane plot for $\theta_i = 65$

Figure 5.6: Incidence plane BRDF plots of a red highly specular plastic. Note the finely resolved specular highlight.

We seek a procedure to evaluate f_r at arbitrary locations at which we do not already have samples. In

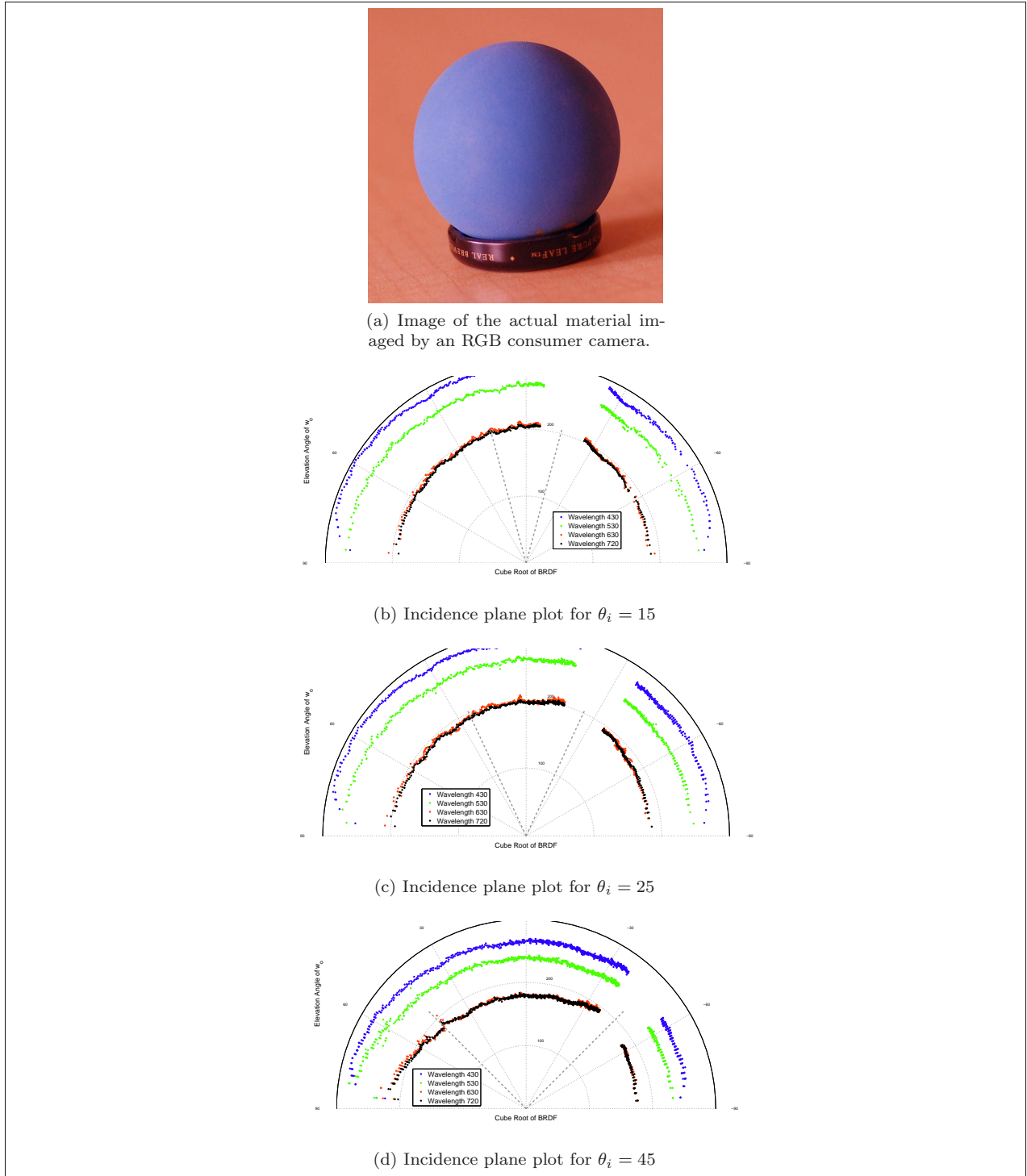


Figure 5.7: Incidence plane BRDF plots of a blue racquetball. Note that this data came from a very early version of our scanning system. At this point, we had to stop measurement very far from retro-reflection due to a large light source housing potentially colliding with the camera. This is why there are “holes” near $\hat{\omega}_i$.

general, this type of problem is known as a *scattered data interpolation problem*[32]. However, our particular

¹*Interpolation* is somewhat of a misnomer. Interpolation typically implies that the approximating function matches the sample values exactly. In scattered data interpolation this restriction is often relaxed and we instead seek a smooth function that fits the sample points over the domain in question.

situation does not allow us to blindly apply techniques from the scattered data interpolation literature. First, we have a very large number of measurements. Our BRDF measurement system is capable of producing up to 30 million samples for a single BRDF. Many unstructured data interpolation algorithms inspect all unstructured measurements to do a single interpolation, and thus are $\mathcal{O}(N)$. This is unacceptable since we need to reconstruct each BRDF at just under 3 million grid points to get our data into the format of Matusik et. al[4]. A second issue is that Euclidean distance between BRDF angular sample positions does not work well for reconstruction.

To avoid the first issue of running time, we use a spatial data structure, the KD-Tree, to speed up our queries when evaluating a BRDF at a new angular position. In our case, the kd-tree is 3 dimensional, since there are 3 angular coordinates in the isotropic BRDF we are interpolating. KD-Trees allow efficient nearest neighbor queries(kNN), and this allows us to speed up our search. For worst case complexity for finding the k-nearest neighbors in a D-dimensional KD-Tree is $\mathcal{O}(kDN^{1-\frac{1}{D}})$. In our situation, $D = 3$ so this algorithm is fast. However, in very high dimensional spaces KD-Trees perform poorly due to the curse of dimensionality.

We also require a function $M : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that takes an angular sample $(\theta_i, \theta_o, \phi_{\text{diff}})$ and returns a 3-vector. This function is designed such that Euclidean distances between samples(after M has been applied) corresponds to samples that are nearby in the BRDF domain. This allows us to make k-nearest neighbor queries into our KD-Tree in order to find k nearby samples to interpolate from. Choices for this function are discussed in the next section. First, we provide pseudocode for our reconstruction algorithm:

Algorithm *3D_BRDF_Resample*(M, k)

- 1.
- (* Using N unstructured BRDF samples, output grid of B brdf measurements *)
2. **for** $j \leftarrow 1$ **to** N
 - (* Add unstructured samples to KD-Tree *)
3. $a = M(\theta_{i,j}, \theta_{o,j}, \phi_{\text{diff},j})$
4. kd_tree.add_sample(a, b_j)
5. **for** $i \leftarrow 1$ **to** B
 - (* Evaluate the BRDF at B grid points *)
6. $(\theta_i, \theta_o, \phi_{\text{diff}}) = \text{get_bin_angular_pos}(i)$
7. $a = M(\theta_i, \theta_o, \phi_{\text{diff}})$
8. $L = \text{kd_tree_knn}(a, k)$
9. $b = \text{unstructured_data_interpolation}(L)$
10. grid[i] = b
11. **return** grid

5.3.1 Distance Metrics for Unstructured BRDF Samples

When resampling the BRDF at a particular position $(\theta_i, \theta_o, \phi_{\text{diff}})$, we first find K BRDF samples nearby this point. However, defining “nearby” for BRDF measurements is a challenging task. Formally, we seek a continuous function $M : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that satisfies the following characteristics:

$$M : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \quad (5.4)$$

$$\|M(s_1) - M(s_2)\|_2 < \epsilon \text{ for nearby samples } s_1 \text{ and } s_2 \quad (5.5)$$

$$\|M(\theta_i, \theta_o, \phi_{\text{diff}}) - M(\theta_i, \theta_o, \phi_{\text{diff}} + i\pi)\|_2 = 0 \forall i \in \mathbb{Z} \text{ (The mapping should reflect isotropy)} \quad (5.6)$$

$$\|M(\theta_i, \theta_o, \phi_{\text{diff}}) - M(\theta_o, \theta_i, \phi_{\text{diff}})\|_2 = 0 \text{ (The mapping should reflect Helmholtz reciprocity)} \quad (5.7)$$

One choice for M was provided by Marschner in his thesis[33]. He provides the following mapping M_m :

$$M_m(\theta_i, \theta_o, \phi_{\text{diff}}) = \left(\sin(\theta_i) \sin(\theta_o), \frac{\phi_{\text{diff}}}{\pi}, \cos(\theta_i) \cos(\theta_o) \right) \quad (5.8)$$

This mapping has the benefit that it effectively doubles the sampling rate by enforcing Helmholtz reciprocity. We also consider two other sampler mappings for M . One mapping, M_s simply consists of using the “standard coordinates” themselves, once we have made sure ϕ_{diff} is normalized to the range $[0, 180]$. A third mapping, M_h , consists of transforming $(\theta_i, \theta_o, \phi_{\text{diff}})$ into halfangle-difference coordinates. The equations for this transformation are included in the work of Rusinkiewicz[31].

5.3.2 Reconstruction Methods for Unstructured BRDF Samples

For this discussion, we can ignore the wavelength dependence of the BRDF. This is because at each angular BRDF sampling position we have a full set of spectral samples. Therefore, whatever procedures we develop for resampling the BRDF can be applied to each spectral band independently. The output from our previous BRDF measurement stage is a set of N BRDF measurements b_1, b_2, \dots, b_N at corresponding angular configurations $p_1 = (\theta_{i1}, \theta_{o1}, \phi_{\text{diff}1}), \dots, p_N = (\theta_{iN}, \theta_{oN}, \phi_{\text{diff}N})$. We can apply a mapping function M to get a set of N points in \mathbb{R}^3 . Using these points, we lookup k nearby samples in our KD-Tree in order to evaluate the BRDF. However, once we have k samples we need a way of combining them into a final measurement. We evaluate two techniques. For the first technique, we compute a simple average of the k nearby samples. Our second technique uses a weighting scheme to favor nearby samples. This weighting scheme is given in the following equations:

$$f(\vec{x}) = \frac{\sum_{i=1}^K b_i w_i}{\sum_{i=1}^K w_i} \quad (5.9)$$

$$w_i = \frac{1}{\|M(\vec{x}), M(\vec{x}_i)\|^p} \quad (5.10)$$

These equations are derived from a method called Shepard Interpolation[32]. Altering the value of p allows the user to control the amount of smoothing in the reconstruction. In the evaluation below, we only consider the simple case of $p = 1$.

5.4 Evaluation of BRDF Resampling

We have outlined several choices for how to reconstruct a BRDF from unstructured samples. In order to determine which reconstruction methods and parameter settings to use, we tested our system using synthetic data. This synthetic data was generated by evaluating an analytical BRDF model at a set of predefined sampling locations. We used the sampling locations generated from an actual BRDF scan using our measurement system. This test allows us to decouple the aspect of the quality of our data versus the reconstruction algorithm being used. For this evaluation, we used RGB BRDFs for the sake of speed since the spectral aspect of our system plays no role in the reconstruction algorithm. The analytic BRDF models we chose were Torrance-Sparrow[34] and Ward[2]. Torrance-Sparrow was used to simulate glossy BRDFs with a Fresnel component, while Ward was used to simulate more specular materials. Note that we used the simpler isotropic ward model shown in Walter's tech report[35]($\alpha_x = \alpha_y$).

5.4.1 Torrance Sparrow BRDF Results

The Torrance-Sparrow BRDF is a microfacet BRDF model, originally developed by Torrance and Sparrow[34], and first introduced to computer graphics by Cook and Torrance[19]. This model allows us to model glossy surfaces. The equations comprising the Torrance-Sparrow model are shown below in equation 5.11.

$$\begin{aligned} f_r &= \frac{DGF}{\pi \cos(\theta_i) \cos(\theta_o)} \\ D &= \frac{e^{\frac{-(\tan^2(\theta_h))}{m^2}}}{4m^2 \cos^4(\theta_h)} \\ G &= \min \left\{ 1, \frac{2 \cos(\theta_h) \cos(\theta_i)}{\cos(\theta_d)}, \frac{2 \cos(\theta_h) \cos(\theta_o)}{\cos(\theta_d)} \right\} \\ F &= \frac{1}{2} \left[\left(\frac{\sin(\theta_t - \theta_d)}{\sin(\theta_t + \theta_d)} \right)^2 + \left(\frac{\tan(\theta_d - \theta_t)}{\tan(\theta_d + \theta_t)} \right)^2 \right] \\ \theta_t &= \sin^{-1} \left(\frac{\sin(\theta_d)}{\eta} \right) \end{aligned} \quad (5.11)$$

In the above equations η is the index of refraction for the material in question. The F term is the Fresnel term, which is the solution of Maxwell's equations at the interface of the material assuming that the material exists in a vacuum($\eta_{\text{env}} = 1$). Here F is the sum of both the solutions for light polarized parallel to the incidence plane and perpendicular[36]. The Fresnel term accounts for reflectance increasing at near grazing configurations. The terms

D and G are the microfacet term, and the self-shadowing term, respectively. Note that the parameter m is the microfacet slope.

Ground truth results for the Torrance-Sparrow model are shown in figure 5.8. The image shown in the ground truth render corresponds to evaluating the analytic Torrance-Sparrow model whenever we evaluate the BRDF in our rendering system. For simplicity, we rendered an RGB image with the same BRDF parameters for each color channel.

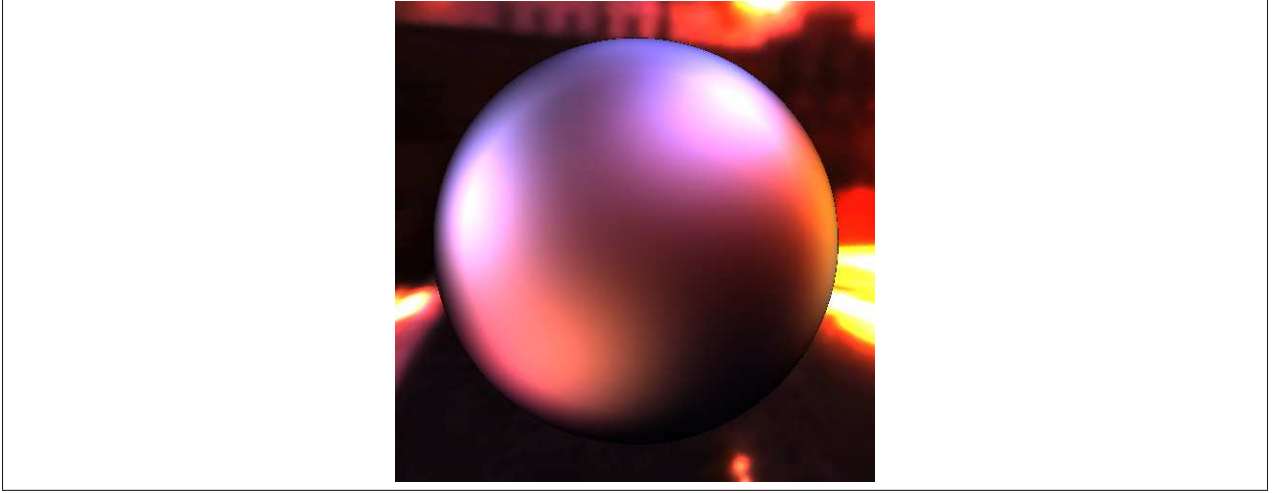


Figure 5.8: Ground truth render for Torrance-Sparrow BRDF. Compare to figures 5.9, 5.10, 5.11, and 5.12. The model parameters have been set to have a smooth, glossy, highlight, as well as a slight Fresnel term.

To generate synthetic, unstructured BRDF data, we used a set of samples originally from an actual scan. We loop over each of the unstructured samples from the scan, and evaluate Torrance-Sparrow with the same parameters shown in image 5.8. This unstructured data is then used in several reconstruction algorithms, with various parameter settings. The resulting HDR images are also tone-mapped with the same parameters. The results are shown in figures 5.9, 5.10, 5.11, and 5.12.

In the above plots, we see the average per-pixel L_1 error in the final column. Unfortunately, this number does not correspond with that a human observer might describe as a “good match”

5.4.2 Ward BRDF Results

To evaluate our system, as well as choose parameters for reconstruction algorithms, we need to evaluate using a specular BRDF as well. For this evaluation, we use the Ward BRDF[2]. Since we deal only with isotropic materials, we use the simplified isotropic described in Walter’s tech report[35]. The isotropic Ward BRDF we use is given by the following formula:

$$f_r(\hat{\omega}_i, \hat{\omega}_o) = \frac{\rho_s}{4\pi\alpha^2 \sqrt{\cos(\theta_i) \cos(\theta_o)}} e^{-\tan^2 \theta_h \left(\frac{\cos^2(\phi_h)}{\alpha^2} + \frac{\sin^2(\phi_h)}{\alpha^2} \right)} \quad (5.12)$$


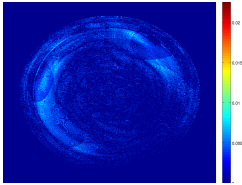

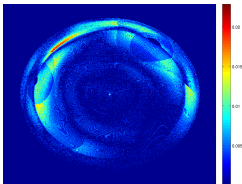

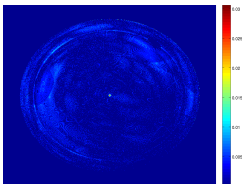
Torrance Sparrow BRDF - Comparison of Reconstruction Parameters - Nearest Neighbor($k = 1$)			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h			0.000715
M_m			0.001501
M_s			0.000692

Figure 5.9: Comparison of BRDF reconstruction parameters for reconstructing a Torrance-Sparrow BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic Torrance-Sparrow model. For these results, a nearest neighbor search($k = 1$) was used.

5.4.3 Parameter Choices for BRDF Reconstruction

After looking at our synthetic tests, we found that $k > 128$ caused excessive smoothing in the results. We also found that a simple average, as opposed to a weighted average, caused excessive smoothing as well. For our synthetic data, each mapping function gave similar results.

To evaluate our choices of reconstruction parameters, we also visualized incidence plane slices of the BRDF(see figure [?]) from a highly specular material. The material is red plastic, shown previously in figure 5.6a. This is a challenging BRDF to resample due to the delta-like nature of the sharp specular highlight. Here we found, that values of $k > 64$ smooth out the highlight excessively. However, values of $k < 16$ are unable to resolve the highlight accurately. We decided that $k = 32$ works best for this type of highly specular material when reconstructing from noisy data. Also, we found that a weighted average was necessary here. The non-weighted scheme smooths out the highlight when using a non-weighted average. We also found that, for this dataset, the “standard” mapping function worked the best. The M_M mapping takes advantage of Helmholtz reciprocity. However, this works to our disadvantage since we have noisy data. If the highlight is off by a fraction of a degree, when we swap $\hat{\omega}_o$ and $\hat{\omega}_i$, this produces the effect of smoothing sharp highlights like the one shown. For these reasons, we settled on mapping M_s , $k = 32$,

and a weighted average. An incidence plane slice using these parameters is shown in figure 5.17. This represents a challenging BRDF to reconstruct due to the sharp highlight. We see that our reconstruction fits the points well.


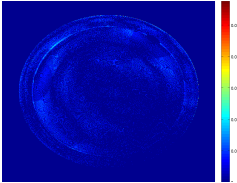

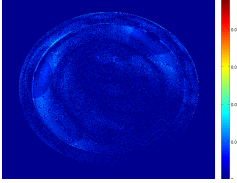

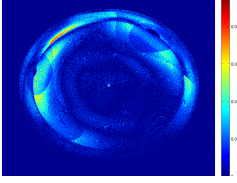
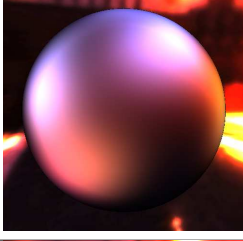
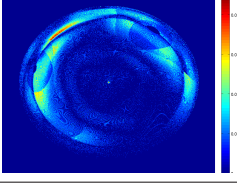
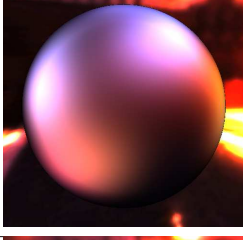
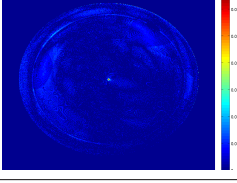

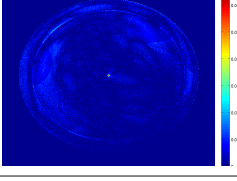
Torrance Sparrow BRDF - Comparison of Reconstruction Parameters - $k = 16$			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h , Avg.			0.000851
M_h , Weighted Avg.			0.000564
M_m , Avg.			0.001435
M_m , Weighted Avg.			0.001432
M_s , Avg.			0.000662
M_s , Weighted Avg.			0.000654

Figure 5.10: Comparison of BRDF reconstruction parameters for reconstructing a Torrance-Sparrow BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic Torrance-Sparrow model. Note that “Avg” refers to a simple mean of all k nearby samples, whereas “Weighted Avg” refers to using equation 5.9 with $p = 1$. For these results, a nearest neighbor search($k = 16$) was used.


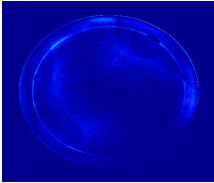

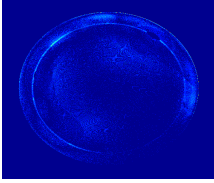

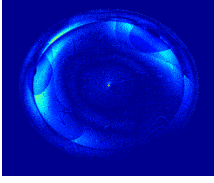
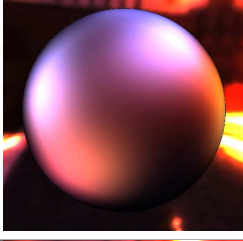
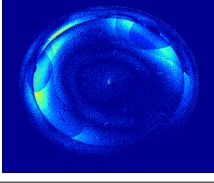
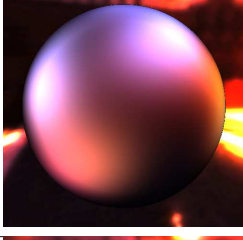
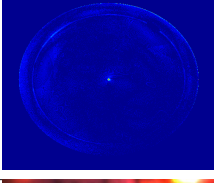

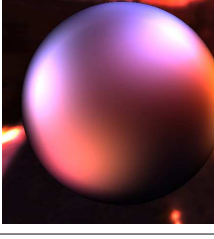
Torrance Sparrow BRDF - Comparison of Reconstruction Parameters - $k = 64$			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h , Avg.			0.000993
M_h , Weighted Avg.			0.000636
M_m , Avg.			0.001317
M_m , Weighted Avg.			0.001293
M_s , Avg.			0.000666
M_s , Weighted Avg.			0.000612

Figure 5.11: Comparison of BRDF reconstruction parameters for reconstructing a Torrance-Sparrow BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic Torrance-Sparrow model. Note that “Avg” refers to a simple mean of all k nearby samples, whereas “Weighted Avg” refers to using equation 5.9 with $p = 1$. For these results, a nearest neighbor search($k = 64$) was used.


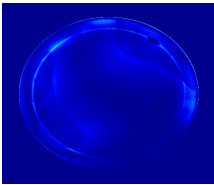

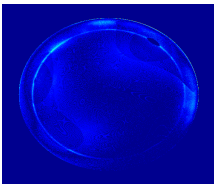

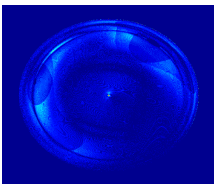

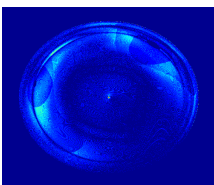

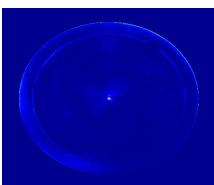

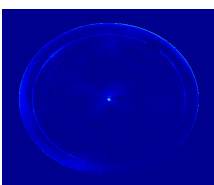
Torrance Sparrow BRDF - Comparison of Reconstruction Parameters - $k = 256$			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h , Avg.			0.002288
M_h , Weighted Avg.			0.001524
M_m , Avg.			0.001688
M_m , Weighted Avg.			0.001542
M_s , Avg.			0.000908
M_s , Weighted Avg.			0.000617

Figure 5.12: Comparison of BRDF reconstruction parameters for reconstructing a Torrance-Sparrow BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic Torrance-Sparrow model. Note that “Avg” refers to a simple mean of all k nearby samples, whereas “Weighted Avg” refers to using equation 5.9 with $p = 1$. For these results, a nearest neighbor search($k = 256$) was used.



Figure 5.13: Ground truth render for isotropic Ward BRDF. Compare to figures 5.14, 5.15, and 5.16. The model parameters have been set to have a large specular highlight and 0 diffuse reflectance. While this is physically implausible, it allows us to test our reconstruction of sharp specular highlights.


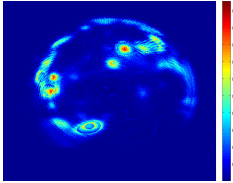

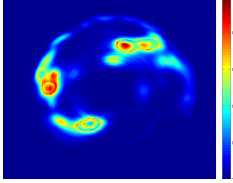

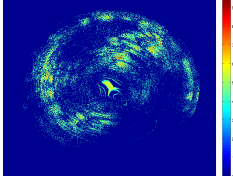
Isotropic Ward BRDF - Comparison of Reconstruction Parameters - Nearest Neighbor($k = 1$)			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h			0.002207
M_m			0.014700
M_s			0.000455

Figure 5.14: Comparison of BRDF reconstruction parameters for reconstructing an isotropic Ward BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic isotropic Ward model. For these results, a nearest neighbor search($k = 1$) was used.


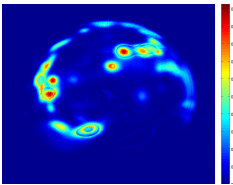

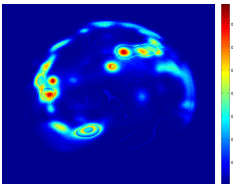

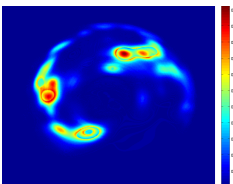

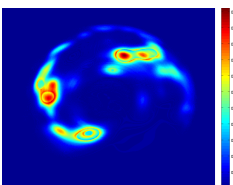

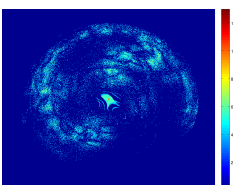
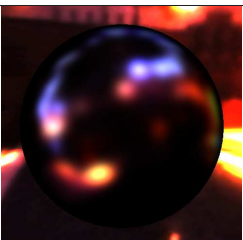
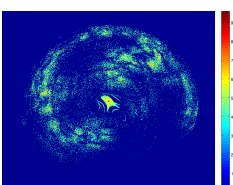
Isotropic Ward BRDF - Comparison of Reconstruction Parameters - $k = 16$			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h , Avg.			0.005284
M_h , Weighted Avg.			0.003982
M_m , Avg.			0.013192
M_m , Weighted Avg.			0.013236
M_s , Avg.			0.000387
M_s , Weighted Avg.			0.000369

Figure 5.15: Comparison of BRDF reconstruction parameters for reconstructing an isotropic Ward BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic isotropic Ward model. Note that “Avg” refers to a simple mean of all k nearby samples, whereas “Weighted Avg” refers to using equation 5.9 with $p = 1$. For these results, $k = 16$ was used for the nearest neighbor search.


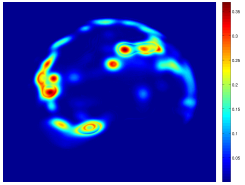

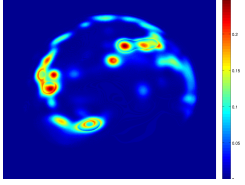
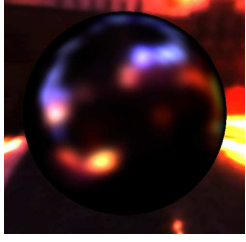
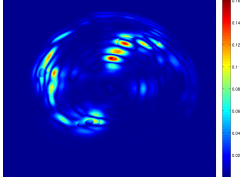
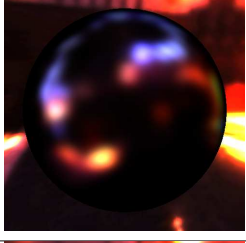
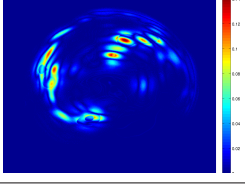

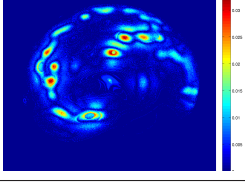

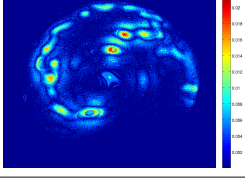
Isotropic Ward BRDF - Comparison of Reconstruction Parameters - $k = 256$			
Parameters	Rendering	L_1 error per-pixel	Avg. L_1 Error
M_h , Avg.			0.021284
M_h , Weighted Avg.			0.013984
M_m , Avg.			0.005764
M_m , Weighted Avg.			0.005087
M_s , Avg.			0.001702
M_s , Weighted Avg.			0.001090

Figure 5.16: Comparison of BRDF reconstruction parameters for reconstructing an isotropic Ward BRDF from unstructured samples. Sample positions were taken from an actual BRDF scan, BRDF values are generated from an analytic isotropic Ward model. Note that “Avg” refers to a simple mean of all k nearby samples, whereas “Weighted Avg” refers to using equation 5.9 with $p = 1$. For these results, $k = 256$ was used for the nearest neighbor search.

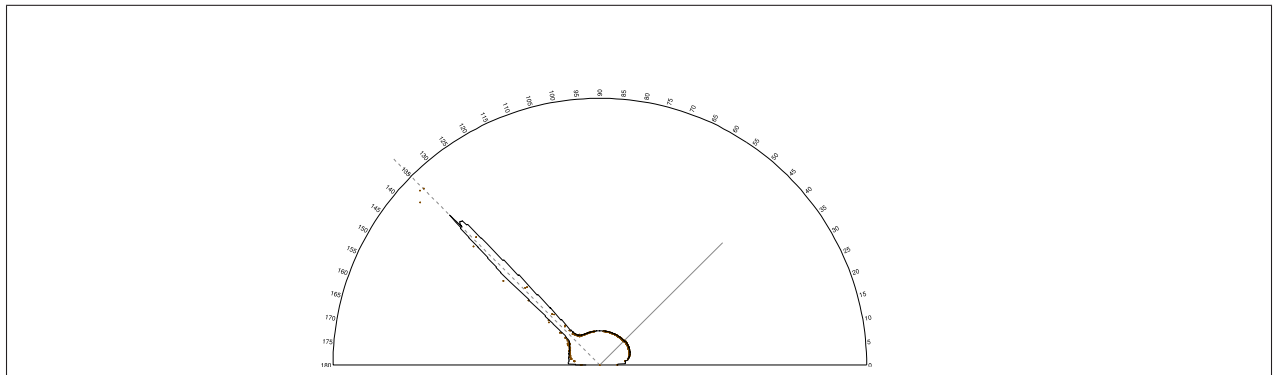


Figure 5.17: Incidence plane plot of specular red plastic unstructured data with binned data overlaid. In this plot, $\lambda = 690 \text{ nm}$. The “standard” mapping was used, along with a weighted average for reconstruction and $k = 32$.

Chapter 6

BRDF Measurement Results

Using our system, we have measured the BRDFs of several materials. Our measurements include BRDFs that are nearly Lambertian, glossy BRDFs, and highly specular BRDFs. We plan on releasing our measurements publicly in the near future. Similar to Ngan et al.[30], we render spheres under an environment map to verify our BRDFs. To facilitate comparison, we use the same environment map as Ngan et al., and have attempted to place the camera in a similar location. However, the environment map we use (the Grace Cathedral environment map) is only RGB. In order to create a multispectral environment map, we use the procedure outlined in appendix A. A few renders of spheres under spectral environment maps are shown below in figure 6.1.

To render our images, we use a multispectral direct-lighting raytracer that runs on the GPU. This raytracer uses a deterministic sampling pattern (as opposed to a Monte Carlo approach). We chose this deterministic sampling approach since our older GPU does not have random number generation easily available. While this removes Monte Carlo noise, it introduces banding artifacts which are visible near the edges of the spheres.

We also rendered images of our materials on some more interesting models. In figure 6.2, we show some of our BRDF data applied to the Stanford dragon model.



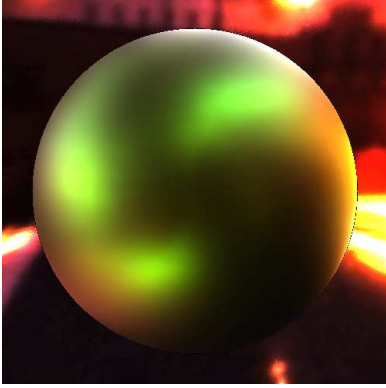

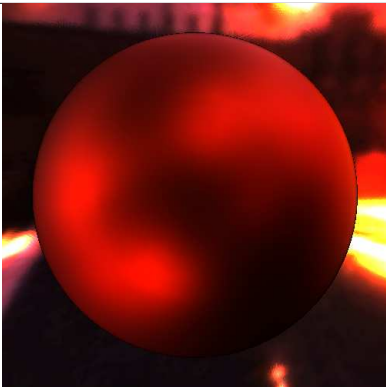

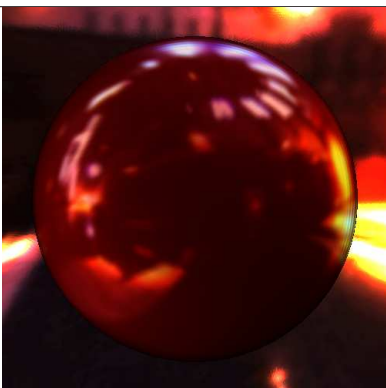

Material	Rendering	Photo of Target used for Measurement
Blue rubber		
Green glossy Christmas ornament		
Red glossy Christmas ornament		
Red specular plastic		

Figure 6.1: Renderings of spheres with our measured spectral BRDFs. In column 1, we have a description of the type of material that was measured. In column 3 we have an photograph, taken with an RGB consumer camera, of the spherical sample that we measured the BRDF from. In column 2, we have a rendering of the BRDF on a sphere in a virtual environment. Direct color comparisons from column 3 to column 2 are not meaningful since our consumer camera performs auto-whitebalancing, and the incident light spectra is unknown in column 3. The tone mapping strategy also affects the final color.

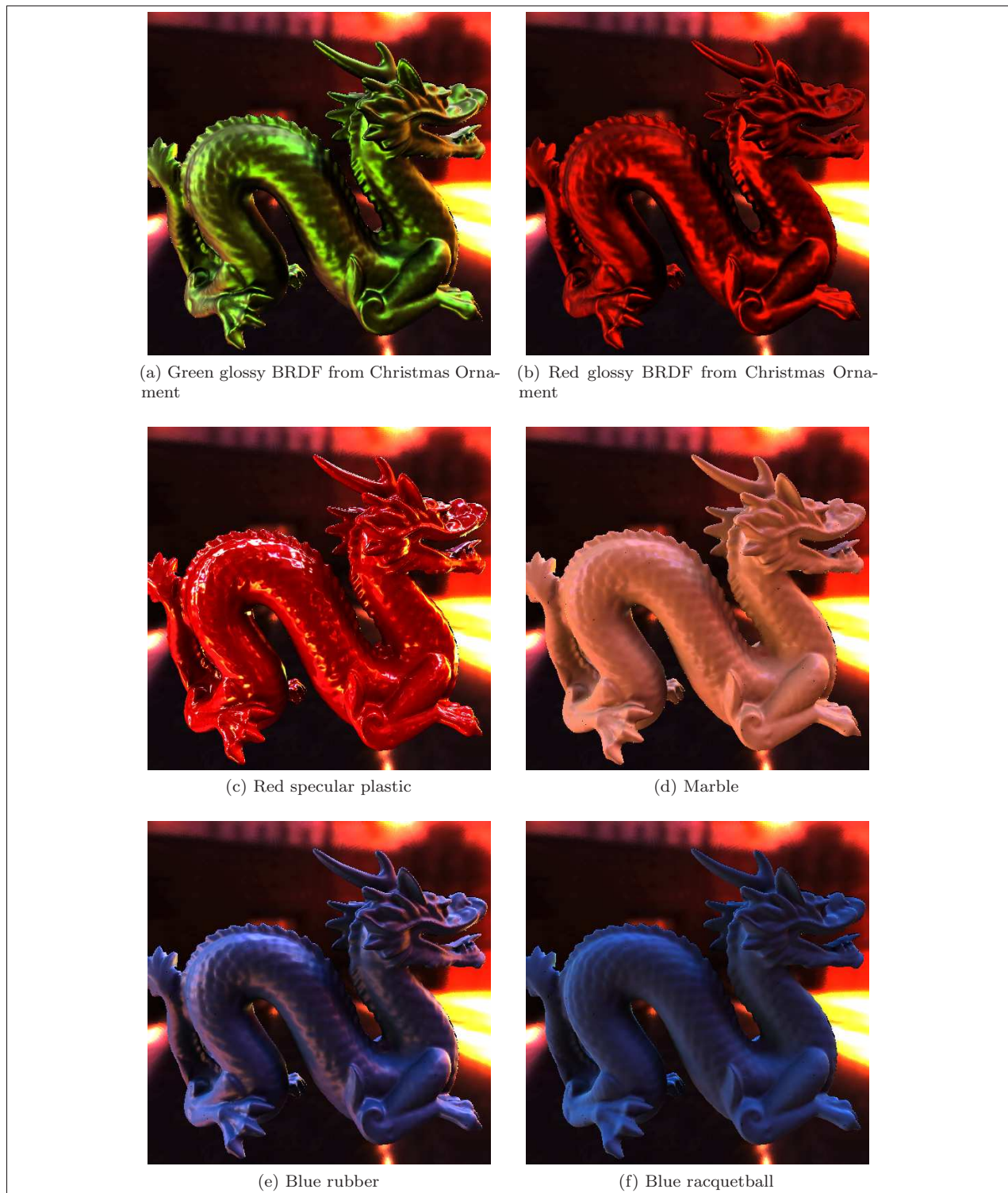


Figure 6.2: Renderings of several multispectral BRDFs applied to the Stanford Dragon Model. The few bright pixels are due to incorrect ray self intersections. The corresponding physical targets here are shown in figure 6.1.

Chapter 7

Multispectral Image Analysis

7.1 Introduction to Multispectral Imaging

Recent advances in sensor technology have made both multispectral and hyperspectral imaging more affordable and simple. This has led to numerous published databases of multispectral(or hyperspectral¹) scenes[7][37][38]. Large databases of spectral images allow us to compute various statistical properties of real world spectra. If we assume that our spectral image database is a reasonable sample of the type of data we expect to encounter in our application, the results from analyzing a spectral image database can influence the design of spectral imaging systems.

Mathematically, the set of all spectra form a vector space. This vector space is the set of all functions $f : [380, 780] \rightarrow \mathbb{R}^+$. As there are an infinite number of such functions f , this vector space is infinite dimensional. Many applications are concerned with measuring or storing the reflectance spectra of objects. Reflectance spectra are often described as smooth, particularly when compared to light emission spectra. In fact, experimental evidence has long shown that reflectance spectra lie near a low dimensional linear subspace(a hyperplane)[8]. The dimensionality of this hyperplane is typically reported to be between 4 and 8, depending on the input data and the standards by which the experimenters define “close.”

The inherent low dimensionality of real world reflectance spectra has profound implications for data compression, spectral rendering, and sensor design. First, if real world spectra lie near a k dimensional linear subspace we can store an approximation of any spectrum by storing k numbers. This compression is useful when storing large amounts of multispectral image data, or when storing spectral data in a memory constrained environment, such as on the GPU. Spectral rendering systems have also been designed to take advantage of the low dimensional nature of reflectance spectra. Peercy[39] designed a rendering system in which spectra are represented by a projection onto k basis functions, as opposed to by point samples. This has the benefit that space is saved, but the process of reflection

¹In general, the term hyperspectral indicates that there are more spectral bands than multispectral. However, what constitutes hyperspectral vs multispectral varies depending on the field of study. Computer vision, remote sensing, and optics practitioners all use these terms in different ways.

now requires a full matrix multiply as opposed to a point-wise multiplication². Furthermore, the low dimensional nature of reflectance can be exploited in spectral imaging systems. For example, Park et. al.[28] created a system for acquiring real-time multispectral video based on illuminating a scene with k rapidly changing basis illuminants. By synchronizing a camera with this rapidly changing light source, one can measure the projection of the underlying reflectance spectra onto these k illuminant basis spectra.

7.2 A Database of Multispectral Images

We conducted an analysis of the database of multispectral images capured by Chakrabarti and Zickler [7]. The images were taken of natural scenes, and do not include human subjects. Areas with human subjects and other imperfections(dust, motion, etc.) were masked out of the data manually. An image from this dataset, converted to sRGB for display, is shown in figure 7.1. The areas which have been masked out are shown in red.



Figure 7.1: Example multispectral image from Chakrabarti et. al.'s database[7], converted to sRGB. Areas masked out are overlaid in with red boxes.

In this dataset 31 spectral bands are present in each image, ranging from 420 nm to 720 nm in 10 nm increments. Considering each pixel's spectral data as a vector $\mathbf{v} \in \mathbb{R}^{31}$, we can analyze the data using linear algebra to answer questions about optimal spectral sensors and compression of multispectral images. After removing masked regions from the spectral image data, we are left with N non-masked pixels, corresponding to N spectra measured from the natural world. Our analysis was conducted on $N \approx 59$ million spectra. We denote this set of spectra by the set $S = \{s_1(\lambda) \dots s_N(\lambda)\}$. Note that through the rest of the document, we use the notation $s_i(\lambda)$ to refer to the i th spectrum as a continuous function, and the notation \mathbf{s}_i to refer to the vector in its point-sampled form, where $\mathbf{s}_i \in \mathbb{R}^{31}$.

²Mathematically, when using point samples the act of reflection is still a matrix multiplication, but the matrix is diagonal. Multiplying by a diagonal matrix is $\mathcal{O}(N)$, as compared to the typical $\mathcal{O}(N^3)$ for a general matrix.

7.3 Principal Component Analysis on Multispectral Images

Using the N spectra from Chakrabarti's database, we form an 31 by N *data matrix*, \mathbf{D} . Each spectrum \mathbf{s}_i forms column i in this matrix. There are 31 rows, since each spectrum has been sampled at 31 discrete wavelengths. We also compute the average spectrum, $\bar{\mathbf{s}}$, and subtract this out from each spectrum in D . This process, commonly referred to as “whitening,” is necessary because the covariance for two sets of observations becomes a dot product when both sets of observations have zero mean.

$$\mathbf{D} = \begin{bmatrix} | & \dots & | & \dots & | \\ \mathbf{s}_1 & \dots & \mathbf{s}_i & \dots & \mathbf{s}_N \\ | & \dots & | & \dots & | \end{bmatrix} - \begin{bmatrix} | & \dots & | & \dots & | \\ \bar{\mathbf{s}} & \dots & \bar{\mathbf{s}} & \dots & \bar{\mathbf{s}} \\ | & \dots & | & \dots & | \end{bmatrix} \quad (7.1)$$

The data matrix D is large, and takes up over 13 GB on disk. Our goal is to find the *principal component* spectra of this data (PCA). Principal component analysis computes an ordered basis $\{\beta_1(\lambda), \dots, \beta_{31}(\lambda)\}$. For any k , the first k basis vectors β_1, \dots, β_k form a k dimensional plane that, when the original data is projected down onto this plane, minimize the sum of the L_2 projection errors.

Given the first k principal component spectra β_1, \dots, β_k , and an arbitrary point-sampled spectrum $\mathbf{s} \in \mathbb{R}^{31}$ that we wish to compress, we can represent \mathbf{s} as k scalars c_1, \dots, c_k each given by the following equation:

$$c_i = \sum_{j=1}^{j=31} \beta_{ij} \mathbf{s}_j \quad (7.2)$$

Note that this is a simple dot product, which is analogous to a definite integral for a continuous spectrum. The fact that this is a dot product is a direct result of the fact that PCA forms an orthonormal basis. Given these k values, c_1, \dots, c_k we can reconstruct our original spectrum via the following:

$$\mathbf{s} = \left(\sum_{j=1}^{j=k} c_j \beta_j \right) + \bar{\mathbf{s}} \quad (7.3)$$

There are two typical techniques used to compute PCA. The first technique is singular value decomposition (SVD) of the data matrix. The second technique is through diagonalization of the covariance matrix ($\frac{1}{N} \mathbf{D} \mathbf{D}^T$). PCA can be thought of as re-expressing our original data in a new orthonormal basis in which the majority of data's variance lies along the new basis vectors.

SVD is typically used for PCA since it avoids having to explicitly compute and store the covariance matrix. However, in our case an SVD based solution is not tenable since \mathbf{D} (the data matrix) is so large it must reside on disk. All readily available SVD implementations require the matrix to be stored in-core. Therefore, we compute the covariance matrix,

$$C = \frac{1}{N} \mathbf{D} \mathbf{D}^T \quad (7.4)$$

which has a much more manageable size of 31 by 31. Computing the covariance matrix itself is a non-trivial process due to the large size of our data. Covariance matrix computation takes nearly 3 hours on our data matrix of size 31 by 59 million. This computation runs on all 4 cores of our system, but we are bottlenecked by disk access speed. Our covariance matrix calculation amounts to multiplying a 13 gigabyte matrix by its transpose. However, the transpose is not ever stored explicitly.

Once we have the covariance matrix, we find all of its eigenvalue-eigenvector pairs. We sort each of these Eigenpairs based on the magnitude of each corresponding eigenvalue. This leaves us with a sorted sequence of vectors $\beta_1, \dots, \beta_{31}$. Here β_1 is the basis vector that accounts for the most variance in the data for our set of input S . These vectors form an ordered basis for our data. The first 8 principal components are shown in figure 7.2.

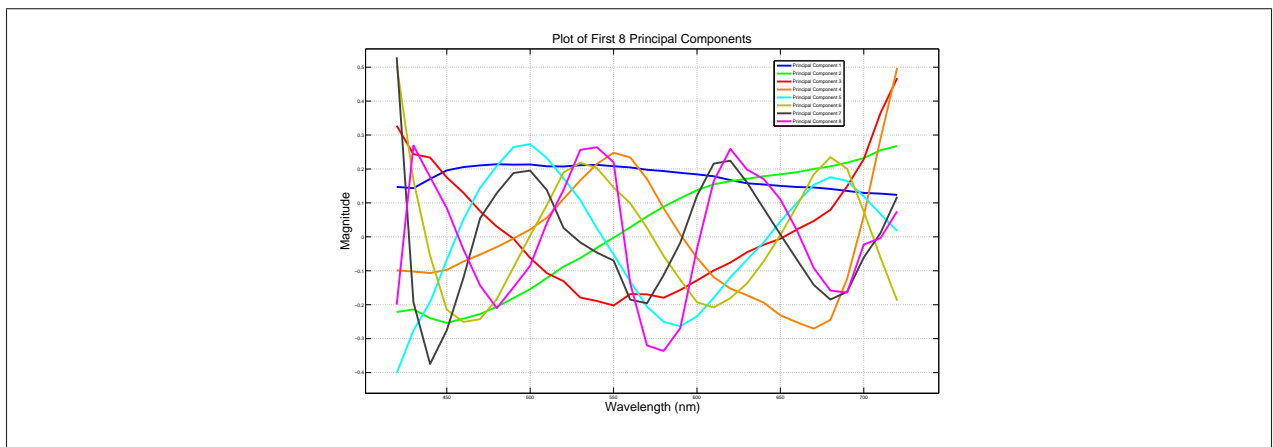


Figure 7.2: First 8 principal components obtained from PCA of 59 million real world reflectance spectra.

Plotting the eigenvalues of these eigenvectors, we see there is a plateau around the 6th eigenvalue (see figure 7.3).

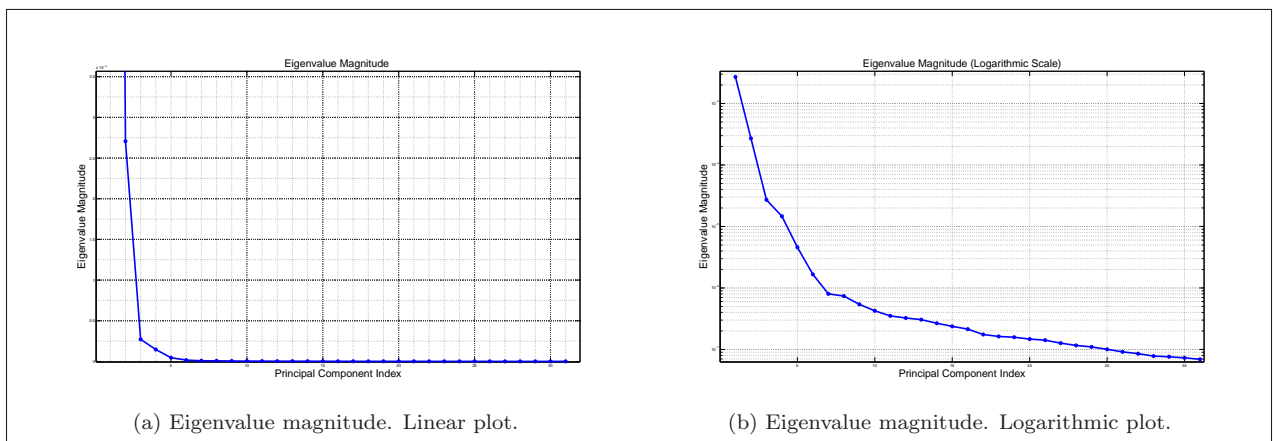


Figure 7.3: Eigenvalue magnitude plot from PCA. Note the plateau starting around the 6th principal component.

The plateau here around 6 indicates that our data is well-represented by 6 principal components. More formally, all of our data lies close to a 6 dimensional linear subspace of \mathbb{R}^{31} . This 6 dimensional hyperplane is spanned by the orthonormal basis $\{\mathbf{b}_1, \dots, \mathbf{b}_6\}$.

This analysis is highly useful for spectral compression. We can project each of our spectra onto the first 6 principal components, and now we only need to store 6 numbers for each spectrum, as opposed to the 31 we stored previously. In figure 7.4 we show an image projected onto various numbers of principal components, and then converted back to RGB for viewing. Without careful inspection, it is difficult to see discrepancies between 6 principal components and 31 principal components(ground truth).

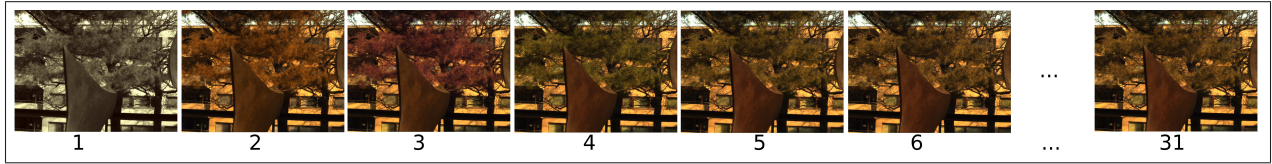


Figure 7.4: Images projected onto principal components, and then converted to RGB. 31 principal components reconstructs the data perfectly, and with 6 principal components the difference is barely noticeable.

7.3.1 Comparision with the Parkkinen Basis

Our principal components agree very well with the Parkkinen basis [8]. The Parkkinen basis was an early PCA based statistical analysis of multispectral data. Parkkenin and his collaborators constructed their analysis on 1257 spectra from a set of photographic color chips. Our analysis is superior in that much more data has been used, and the data is much more representative of the natural world. Perhaps surprisingly, our results are similar to Parkkinen's³. This is evidence that Parkkinen's comparatively small amount of data was indeed sufficient to draw statistical conclusions from. Figure contrasts our principal component spectra with that of Parkinnen et al.

7.4 Non-Negative Matrix Factorization on Multispectral Images

PCA is very useful for data compression. However, PCA is of limited utility when designing spectral sensors. Color imaging systems typically work by recording the definite integrals of the following form:

$$P_i = \int_{\Lambda} E(\lambda) C_i(\lambda) d\lambda \quad (7.5)$$

Here, i indicates the i th color band being recorded by the camera. For an rgb camera, $i \in \{1, 2, 3\}$. $E(\lambda)$ is the incoming SPD, and $C_i(\lambda)$ is the spectral sensitivity at color channel i . Typically, $C_i(\lambda)$ is the product of the spectral transmission of the optics, as well as the quantum efficiency of the CCD or underlying sensor. Ideally we would be able to fabricate devices with arbitrary spectral sensitivities $C_1(\lambda), \dots, C_k(\lambda)$ for each of our k channels. In order

³Note that several of our eigenvectors had to be negated to look visually similar to Parkkinen's. This is not an issue, since $\mathbf{Ax} = \lambda\mathbf{x}$ clearly holds for any $-\mathbf{x}$ as well.

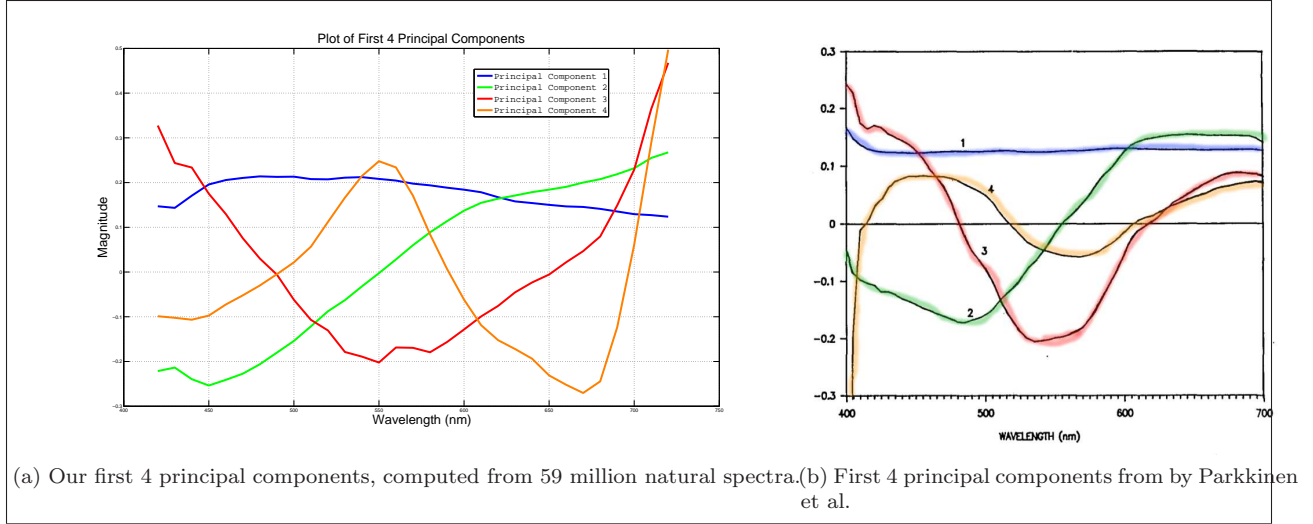


Figure 7.5: Comparison of our PCA results vs that of Parkkinen et al. Our results appears to be qualitatively similar to that of Parkkinen et. al., despite their results only being computed on 1257 spectra. Note that Parkkinen’s data begins at 400nm, whereas our data begins at 420nm. Furthermore, Parkkenin expressed doubts regarding the quality of his data in the near-UV(< 420 nm) wavelengths due to a low signal to noise ratio.

to be physically realizable, spectral sensitivity functions must be non-negative ($C_i : [0, +\infty) \rightarrow [0, +\infty) \forall i$). In multispectral iamging, we want our spectral sensitivities(basis functions) to record $E(\lambda)$ as accurately as possible. We can’t use our basis obtained from PCA here, since the results from PCA can be negative as well as positive.

In order to enforce a non-negativity constraint on our previous result, we need another type of analysis. One suitable tool is *Non-Negative Matrix Factorization*(NMF). In NMF we seek to factor our non-negative data matrix \mathbf{D} into $\mathbf{D} = \mathbf{WH}$ where \mathbf{W} and \mathbf{H} are also non-negative. Here, the user is in control of a parameter k , which corresponds to the number of columns in \mathbf{W} and the number of rows in \mathbf{H} .

Once this factorization is complete, the k columns of \mathbf{W} correspond to a set of strictly positive basis vectors in which our data is well represented. Similar to PCA, we can project onto these k basis vectors and represent our original data. However, since these vectors are positive, they can also be thought of as the transmission spectra of an optical filter or as the QE of a CCD’s response.

One of the downsides of NMF is that it may converge to a solution that is not globally optimal. NMF operates through an iterative ACLS(alternating constrained least squares) algorithm. The algorithm alternates between holding \mathbf{W} fixes and solving for \mathbf{H} , and then holding \mathbf{H} fixed and solving for \mathbf{W} . An initial guess is supplied by the user, which can also affect convergence. In our case, the initial guess used was positive white noise in the range $[0, 1]$. When using NMF we have to be wary that our results might represent a local(as opposed to global) minimum.

Unfortunately, the available NMF implementations require the data matrix \mathbf{D} to reside in memory. Our data matrix is over 13 GB, and thus could not fit into memory. As a result, we were forced to randomly sample the input images to create a data matrix. We ran each NMF experiment 6 times with a different initial guess and different random sample positions. The results were similar each run, indicating that our result is insensitive to

initial conditions and that our sampling is adequate. Our NMF results for various values of k are shown below in figure 7.6.

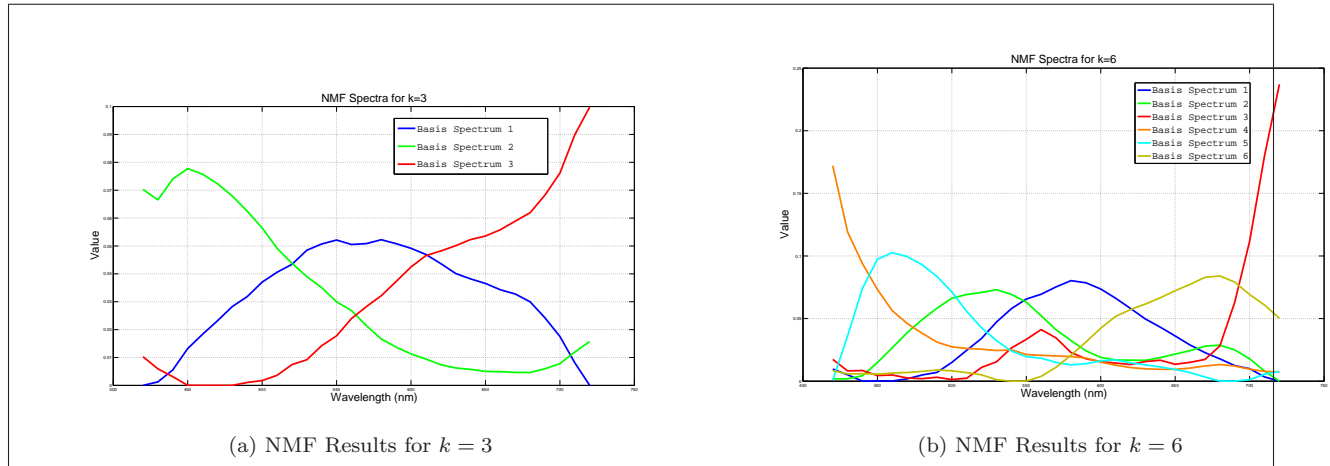


Figure 7.6: Results from performing NMF on 50000 input spectra. Each test was conducted 6 times and similar results were found on each run.

As a simple validation of our NMF results, we took an multispectral image that was excluded from our input data matrix, and projected it onto the NMF basis functions. We then computed the L^2 difference between the reconstructed spectrum after projection, and the underlying ground truth spectrum. This L^2 error, per pixel, is visualized in figure 7.7. We see that NMF does a good job representing spectral data across the majority of the image.



Figure 7.7: Per-pixel L^2 error between the original spectrum and the reconstructed spectrum after projection onto the NMF basis.

7.5 Implications of PCA and NMF Analysis for BRDF Measurement

The results of our analysis presented above have broad implications for BRDF measurement systems. Measured spectral BRDF data is very large. A single 31 channel measured BRDF, where each spectral channel is stored similarly to the MERL format[4] takes up 340 Megabytes of space. This storage requirement is prohibitive on graphics cards, and would only allow us to render using a small amount of BRDFs. Our resulting PCA spectra could be used to compress this data. If we used 6 principal components, which our analysis suggest is adequate, we could reduce storage size for a single BRDF to 170 Megabytes.

Our results from NMF have interesting implications in terms of new spectral BRDF measurement systems. Our existing system uses one camera with a liquid crystal tunable filter(LCTF) in front of the lens. The LCTF allows us to sample the wavelength domain of the BRDF sequentially. However, if we instead used a series of cameras, each with a different spectral filter in front, we could sample the wavelength domain in parallel. We could use narrowband spectral filters, with transmission properties similar to our LCTF. However, narrowband filters limit optical throughput and increase required integration time and noise. Broadband spectral filters don't have these issues. One option would be to manufacture filters with similar transmission properties to those shown in our NMF analysis.

Chapter 8

Conclusion

We presented a new system for multispectral image based BRDF measurement. Our system has been carefully calibrated, and the data generated has already proved useful for rendering. We are able to accurately measure a wide variety of materials, including Lambertian, glossy, and highly specular materials. Our data is some of the first multispectral BRDF data with dense angular sampling in computer graphics community. We plan to release our data publicly in the near future.

We evaluated several different aspects of the calibration process, and compared various reconstruction schemes for the first time. Furthermore, we considered spectral calibration in more depth than previous authors of multispectral BRDF measurement work in computer graphics. We also proposed a novel new calibration technique for determining local per-pixel view and light vectors. This approach is based on solving several non linear optimization problems. This approach is simpler than those presented by previous authors and still yields high-quality results.

Lastly, we conducted a large statistical analysis of a database of multispectral images using Principal Component Analysis(PCA) and Non-Negative Matrix Factorization(NMF). This statistical analysis confirmed the low-dimensional nature of reflectance spectra. Furthermore, our NMF results demonstrated that it may be possible to construct image based BRDF measurement systems using broadband spectral filters. Such a design would allow measurement of the BRDFs spectral component in parallel. We leave the construction of such a system to future work.

Appendix A

Spectral Environment Map Generation

In order to render a images with a spectral BRDF, it is necessary to have an environment for the material to interact with. Previous work on RGB BRDF measurement showed their results using the “Grace Cathedral[40]” environment map[6][30]. We chose to use the same environment map in order to facilitate comparison with previous work and because no multispectral environment maps were readily available. However, the Grace Cathedral environment map has RGB data, not multispectral.

In order to use an RGB environment map in a multispectral renderer, we need a process to convert RGB values to spectra. This problem was previously investigated by Smits[41]. Mathematically, this problem is under-constrained. Consider a situation in which we would like to convert a RGB triple $\mathbf{C} \in \mathbb{R}^3$ to a spectrum $\mathbf{x} \in \mathbb{R}^N$. This can be posed as solving an under-determined linear system as follows.

$$\begin{bmatrix} \cdots & \bar{\mathbf{r}} & \cdots \\ \cdots & \bar{\mathbf{g}} & \cdots \\ \cdots & \bar{\mathbf{b}} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{x} \\ \vdots \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (\text{A.1})$$

$$\mathbf{A}\mathbf{x} = \mathbf{C}$$

Note that in equation A.1, the vectors $\bar{\mathbf{r}}$, $\bar{\mathbf{g}}$, and $\bar{\mathbf{b}}$ are the red, green, and blue color matching functions(see figure 2.6), respectively.

Unfortunately, when we want to create an environment map with $N > 3$ (any multispectral environment map), this linear system is under-determined and has an infinite number of solutions. We add a *regularization* term to this system to create a well-posed problem. We regularize the system by penalizing inflection points in the resulting SPD \mathbf{x} . Typically reflection spectra are smooth, so this solution technique favors smooth spectra. Based on simple finite

differences, the following expression computes the second derivative for an arbitrary function $f(x)$:

$$f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} \quad (\text{A.2})$$

We use this rule, with a weighting parameter α , in our linear system as the regularization term. The larger the value of α , the smoother our solutions becomes. Our regularized system is given by the following linear system:

$$\begin{bmatrix} \dots & \dots & \bar{\mathbf{r}} & \dots & \dots \\ \dots & \dots & \bar{\mathbf{r}} & \dots & \dots \\ \dots & \dots & \bar{\mathbf{r}} & \dots & \dots \\ \alpha & -2\alpha & \alpha & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \alpha & -2\alpha & \alpha \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \mathbf{x} \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{A.3})$$

Using equation A.3, we can create a spectral environment map from an RGB environment map. We simply invert the left hand side matrix in equation A.3, and multiply at by the RGB vector at each pixel in the environment map. However, this still poses issue. Unfortunately, this can produce negative spectra. It is impossible to constrain a linear system to produce non-negative results without employing quadratic programming or a similar tool. However, we found that simple clamping the resulting spectra to 0 in the rare case of negative values does not produce objectionable results. This is an workable solution, since spectral environment map generation is not the focus of our work, and because when re-converted back to RGB the result looks visually quite similar to the original input.

Examples of spectral BRDFs rendered with our spectral environment map are shown in figures 6.1 and figure 6.2. When the environment map of these figures are compared visually to the RGB environment map in 5.8, we see little objectionable visual difference.

Bibliography

- [1] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA, 2001. ACM.
- [2] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, July 1992.
- [3] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, and Kenneth E. Torrance. Image-based bidirectional reflectance distribution function measurement. *Appl. Opt.*, 39(16):2592–2600, Jun 2000.
- [4] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 759–769, New York, NY, USA, 2003. ACM.
- [5] Hongsong Li, Sing Choong Foo, Kenneth E. Torrance, and Stephen H. Westin. Automated three-axis gonio-reflectometer for computer graphics applications. In *Optical Engineering*, page 2006, 2005.
- [6] Wojciech Matusik, Hanspeter Pfister, Matt Br, and Leonard Mcmillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22:759–769, 2003.
- [7] A. Chakrabarti and T. Zickler. Statistics of Real-World Hyperspectral Images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 193–200, 2011.
- [8] J. P. S. Parkkinen, J. Hallikainen, and T. Jaaskelainen. Characteristic spectra of munsell colors. *J. Opt. Soc. Am. A*, 6(2):318–322, Feb 1989.
- [9] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *In NIPS*, pages 556–562. MIT Press, 2000.
- [10] Michael F. Cohen, John R. Wallace, and Hanrahan Pat. *Radiosity and Realistic Image Synthesis*. Academic Press, 1 edition, 1993.
- [11] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.
- [12] Philip Dutre, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination*. A. K. Peters, Ltd., Natick, MA, USA, 2002.

- [13] Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.
- [14] F E Nicodemus, J C Richmond, J J Hsia, I W Ginsberg, and T Limperis. Geometrical considerations and nomenclature for reflectance. *Science And Technology*, 160(October):1–52, 1977.
- [15] Matthias B. Hullin, Johannes Hanika, Boris Ajdin, Hans-Peter Seidel, Jan Kautz, and Hendrik P. A. Lensch. Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions. *ACM Trans. Graph.*, 29(4):97:1–97:7, July 2010.
- [16] Gunter Wyszecki and Stiles W. S. *Color Science Concepts and Methods, Quantitative Data and Formulas*. John Wiley and Sons, Inc., 1 edition, 1967.
- [17] Roy Hall. *Illumination and Color in Computer Generated Imagery*. Springer Verlag, 1 edition, 1988.
- [18] James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, July 1977.
- [19] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982.
- [20] Steven A. Shafer. Color. chapter Using color to separate reflection components, pages 43–51. Jones and Bartlett Publishers, Inc., USA, 1992.
- [21] James E Proctor and P. Yvonne Barnes. Nist high accuracy reference reflectometer-spectrophotometer. *Journal of Research of the National Institutes of Standards and Technology*, 101(5), September 1996.
- [22] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, July 1992.
- [23] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 1 edition, August 2002.
- [24] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, November 2000.
- [25] Reinhard Klein Martin Rump, Arno Zinke. Practical characterization of trichromatic cameras. *Siggraph Asia 2010*, 2010.
- [26] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 31:1–31:10, New York, NY, USA, 2008. ACM.
- [27] Mitchell R. Rosen Francisco H. Imai and Roy S. Berns. Comparative study of metrics for spectral match quality. pages 492–496. Munsell Color Science Laboratory, Rochester Institute of Technology, 2002.
- [28] J. Park, M. Lee, M. D. Grossberg, and S. K. Nayar. Multispectral Imaging Using Multiplexed Illumination. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2007.

- [29] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9:112–147, 1998.
- [30] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental validation of analytical brdf models. In *ACM SIGGRAPH 2004 Sketches*, SIGGRAPH '04, pages 90–, New York, NY, USA, 2004. ACM.
- [31] Szymon Rusinkiewicz. A new change of variables for efficient BRDF representation. In *Rendering Techniques (Proc. Eurographics Workshop on Rendering)*, 1998.
- [32] D.R. Kincaid and E.W. Cheney. *Numerical analysis: mathematics of scientific computing*. Brooks/Cole series in advanced mathematics. Brooks/Cole, 2002.
- [33] S.R. Marschner. *Inverse Rendering for Computer Graphics*. PhD thesis, 1998.
- [34] K. E. TORRANCE and E. M. SPARROW. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.*, 57(9):1105–1112, Sep 1967.
- [35] Bruce Walter. Notes on the ward brdf. Technical report, April 2005.
- [36] Tim Weyrich, Jason Lawrence, Hendrik Lensch, Szymon Rusinkiewicz, and Todd Zickler. Principles of appearance acquisition and representation. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 80:1–80:119, New York, NY, USA, 2008. ACM.
- [37] Sérgio M. C. Nascimento, Flávio P. Ferreira, and David H. Foster. Statistics of spatial cone-excitation ratios in natural scenes. *J. Opt. Soc. Am. A*, 19(8):1484–1490, Aug 2002.
- [38] F. Yasuma, T. Mitsunaga, D. Iso, and S.K. Nayar. Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum. Technical report, Nov 2008.
- [39] Mark S. Peercy. Linear color representations for full speed spectral rendering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 191–198, New York, NY, USA, 1993. ACM.
- [40] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 32:1–32:10, New York, NY, USA, 2008. ACM.
- [41] Brian Smits. An rgb to spectrum conversion for reflectances, 2000.