A Data-Driven Reflectance Model

by

Wojciech Matusik

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2003

Author ....
Department of Electrical Engineering and Computer Science
August 29, 2003

Certified by ....
Leonard McMillan
Associate Professor of Computer Science
University of North Carolina, Chapel Hill
Thesis Supervisor

Accepted by..... ....
Arthur C. Smith
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

A Data-Driven Reflectance Model

by

Wojciech Matusik

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

I present a data-driven model for isotropic bidirectional reflectance distribution functions (BRDFs) based on acquired reflectance data. Instead of using analytic reflectance models, each BRDF is represented as a dense set of measurements. This representation allows interpolation and extrapolation in the space of acquired BRDFs to create new BRDFs. Each acquired BRDF is treated as a single high-dimensional vector taken from the space of all possible BRDFs. Both linear (subspace) and non-linear (manifold) dimensionality reduction tools are applied in an effort to discover a lower-dimensional representation that characterizes the acquired BRDFs. To complete the model, users are provided with the means for defining perceptually meaningful parametrizations that allow them to navigate in the reduced-dimension BRDF space. On the low-dimensional manifold, movement along these directions produces novel, but valid, BRDFs.

By analyzing a large collection of reflectance data, I also derive two novel reflectance sampling procedures that require fewer total measurements than standard uniform sampling approaches. Using densely sampled measurements the general surface reflectance function is analyzed to determine the local signal variation at each point in the function's domain. Wavelet analysis is used to derive a common basis for all of the acquired reflectance functions, as well as a non-uniform sampling pattern that corresponds to all non-zero wavelet coefficients. Second, I show that the reflectance of an arbitrary material can be represented as a linear combination of the surface reflectance functions. Furthermore, this analysis specifies a reduced set of sampling points that permits the robust estimation of the coefficients of this linear combination. These procedures dramatically shorten the acquisition time for isotropic reflectance measurements.

Thesis Supervisor: Leonard McMillan
Title: Associate Professor of Computer Science
University of North Carolina, Chapel Hill

Acknowledgments

I would like to express my gratitude to the following people who helped me during the last five years at MIT.

First and foremost, I would like to thank my advisor Professor Leonard McMillan for guiding me through both the Masters and the Ph.D. programs. Not only did Professor McMillan encourage my interest in computer graphics but he has given incredible support during the key moments of my graduate studies. Professor McMillan's original thoughts and insights were the motivating force in developing these ideas. It was a great pleasure to work with him. I also would like to express my sincere thanks to Hanspeter Pfister for co-advising me on the research project that led to this thesis as well as collaborating on other research during the last three years. His enthusiasm and stimulating discussions were crucial in all of these projects. Both Professor McMillan's and Hanspeter Pfister's deep interest in the project and constant encouragement were crucial in completing the work.

I would like to thank Matt Brand for advising me on many parts of the project. Technical discussions with Matt, his algorithms, and his research code were essential in developing this data-driven reflectance model.

I also would like to express my gratitude to my thesis committee members Professors Fredo Durand and Ted Adelson for their time and their valuable input on the draft of this dissertation. Discussions with Professors Fredo Durand, Steven Gortler, Julie Dorsey, and Markus Gross during various stages of the project were enormously helpful.

Thanks also go to Henrik Wann Jensen for his assistance with his Dali renderer, Paul Lalonde for his wavelet shader, Paul Debevec for the light probes, and Matt Peters for the intial data analysis. I wish to thank Joe Marks for the continual support of this project and for hosting it at MERL. Also, my thanks go to Barb Cutler, Ray Jones, and Addy Ngan for providing me with constructive feedback on the drafts of the dissertation. In addition, I would like to thank the whole Computer Graphics Group at MIT for their support.

Finally, my deepest gratitude goes to my family. I would like to thank my parents and my wife Gosia for their love, patience and encouragement throughout the years.

Contents

List of Figures

Chapter 1

Introduction

Modeling and measuring how light is reflected from surfaces is a central theme in both computer graphics and computer vision. The Bidirectional Reflectance Distribution Function (BRDF) describes reflectance under the assumption that all light transport occurs at a single surface point[1]. A general BRDF describes reflected light as a four-dimensional function of incident and exitant directions. This thesis focuses on the important subclass of *isotropic* BRDFs, for which rotations about the surface normal can be ignored. Isotropic BRDFs are functions of only three angles (the incident illuminations angle relative to the surface normal and two angles to parameterize the reflected radiance).

Traditionally, physically inspired analytic reflectance models [7, 17, 39] or empirical reflectance models [41, 55, 25] provide the BRDFs used in computer graphics and computer vision. These BRDF models are only approximations of reflectance of real materials. Furthermore, most analytic reflectance models are usually limited to describing only particular subclasses of materials – a given reflectance model can represent only the phenomena for which it is designed. The models have evolved over the years to become more complex, incorporating more and more of the underlying physics. It is worth to note that many of the physically based models are based on material parameters that in principle could be measured, but in practice are difficult to acquire.

An alternative to directly measuring model parameters is to directly measure values of

---

[1] While the original definition of BRDF [37] also describes some limited subsurface scattering effects, the light interaction at a single surface point is commonly accepted in the literature.

the BRDF for different combinations of the incoming and outgoing angles and then fit the measured data to a selected analytic model using various optimization techniques [55, 57, 25, 44]. There are several shortcomings to this measure-and-fit approach. First, a BRDF represented by an analytic function is only an approximation of real reflectance – measured values of the BRDF are usually not exactly equal to the values of the analytic model. The measure-and-fit approach is often justified by assuming that there is an inherent noise in the measurement process and that the fitting process filters out these errors. This point of view, however, ignores more significant modelling errors due to approximations made by the analytic reflectance model. Many of the salient and distinctive aspects of a material reflectance properties might lie within the range of these modelling errors. Second, the choice of the error function with which the optimization should be performed is not obvious. For example, error based on the Euclidean distance is a poor metric since it tends to overemphasize the importance of the specular peaks and ignore the off-specular reflection properties. Finally, there is no guarantee that the optimization process will yield the best model. Since most BRDF models are highly non-linear, the optimization framework used in the fitting process relies heavily on the initial guess for the model parameters. The quality of the initial guess can have a dramatic impact on the final parameter values of the model.

The third approach to modelling reflectance is to acquire dense measurements of reflectance and use these tabulated measurements directly as a BRDF. This approach preserves those subtleties of the reflectance function that are lost in a data-fitting approach. The classical device for measuring BRDFs is the gonio-reflectometer [8, 10, 38], which is composed of a photometer and light source that are moved relative to a surface sample under computer control. By design, such devices measure a single radiance value at a time, making this process very time-consuming. There have been efforts to make this acquisition process more efficient by measuring many BRDF samples at once. This can be achieved by using a digital camera and mirrors [55, 11] or spherical samples of the measured material [34]. The approach of using tabulated BRDFs becomes even more expensive if the reflectance for all materials in a scene needs to be measured and stored. Furthermore, one ends up with a collection of measured BRDFs and not with a parameterized reflectance model. Any change to the material property would require finding a real material with the desired property and acquiring its reflectance.

I propose another alternative – a data-driven approach for modelling surface reflectance. I capitalize on the fact that it is feasible to rapidly acquire accurate reflectance measurements. I acquire BRDFs for a large representative set of materials. Materials in my collection include metals, paints, fabrics, minerals, synthetics, organic materials, and others. I introduce a new approach to BRDF modelling, an approach that is data driven – it interpolates and extrapolates new BRDFs from the representative BRDF data. My approach has the advantage that the produced BRDFs look very realistic since they are based on the measured BRDFs. Furthermore, I provide a set of intuitive parameters that allow users to change the properties of the synthesized BRDF. I also let users create their own parameters by labelling a few representative BRDFs. I believe that this way of specifying model parameters makes my model much easier to use and control than the analytic models in which the meaning of parameters is often non-intuitive [40].

In my data-driven model, it is undesirable to store all acquired BRDFs explicitly. This leads me to the analysis of the space of all possible BRDFs for common materials in the world. A BRDF for these materials is not an arbitrary function, and I seek a representation for all possible functions corresponding to physical BRDFs. I treat each of my acquired BRDFs as a single high-dimensional vector where each measurement is an element of this vector. Then I apply both linear and non-linear dimensionality reduction tools to obtain a low dimensional manifold along with its linear embedding space that characterizes the set of BRDFs I measured. In the process I also obtain a backprojection operator that maps the embedding space of the manifold to the original BRDF space. Therefore, I can always compute the corresponding BRDF for each point on the manifold. An interesting side effect of my approach is that it suggests an inherent dimensionality for the space of all isotropic BRDFs.

The measurement and analysis of a relatively large collection of densely sampled isotropic BRDFs from many different materials also lead to more efficient measurement procedures of reflectance functions. I observe that sampling BRDF uniformly is very inefficient since it requires a huge amount of measurements. For example, an angular resolution of 0.5°

requires more than 46 million measurements. Therefore, I start by answering the question, “What is the required sampling frequency over the domain of the isotropic BRDF to adequately measure it?” The first proposed measurement procedure is based on the wavelet analysis of the space of measured BRDFs. I observe that the BRDFs in my dataset have varying frequency content at various points in their domain. For example, specular highlights have complicated local spectrums that contain high frequencies, whereas off-specular signals are typically smooth with simple local spectrums. I exploit these properties of BRDF spectrums to derive an efficient measurement procedure that employs a non-uniform sampling of the reflectance function. The sampling density at each point of the function domain is a function of the signal frequency that adequately represents any BRDF.

The linear reflectance model leads to another efficient measurement procedure. I note that the set of the basis functions generated by the linear model is the optimal basis to represent the BRDFs in the original dataset. Furthermore, I show that new isotropic BRDFs (not in the original set) can also be represented as linear combinations of these basis functions or as linear combinations of the BRDFs in the original set. This implies that one needs to make very few measurements, just enough to constrain the few parameters of the linear model, in order to obtain the densely sampled representation of any new BRDF.

To summarize, the central thesis of my dissertation can be expressed in the following statement:

Reflectance of materials in nature can be modelled as a linear combination of a small set of basis functions derived from analyzing a large number of densely sampled reflectance functions of different materials.

In more detail, the main contributions of my thesis are as follows:

- I introduce the first data-driven reflectance model.
- I introduce a set of perceptually-based parameters for this model. I also let users create their own parameters.
- I analyze both linear and non-linear dimensionality of the space of isotropic BRDFs.
- In my model the parameter values are pre-defined for many typical materials – the

materials I have measured. Using my model I can also generate difficult to represent effects such as rust, oxidation, or dust.

- I demonstrate two novel BRDF sampling procedures. The procedures require few measurements and reconstruct dense BRDF representation to within measurement error. The first procedure is based on the wavelet analysis of BRDFs in my dataset. The second procedure follows directly from subspace analysis of BRDFs in my dataset.

1.1 Thesis Overview

The rest of my thesis is organized as follows: In the second chapter I present background material about reflectance and describe the most significant previous work in the reflectance modelling and representations. I also outline linear and non-linear dimensionality reduction techniques that are used to construct my data-driven reflectance model. In the third chapter I describe the measurement system and procedure that are needed to obtain densely sampled, tabulated BRDFs for a large collection of different materials. In chapter 4, I present the linear and non-linear analysis of my BRDF dataset. Chapter 5 contains the description of the methods used to create user-defined, perceptual parameterizations for my data-driven model. The subject of chapter 6 is efficient measurement and storage; I show how to represent the data-driven reflectance model using wavelets. The wavelet representation and the linear analysis of chapter 4 lead to the efficient and simple measurement procedures. These procedures allow us to measure BRDFs using much fewer samples than standard procedures. Finally, chapter 8 summarizes the results and discusses directions for future work.

Chapter 2

Previous Work

There has been a great deal of research on how light reflects from surfaces. Many different aspects of reflectance have been examined. In this chapter, I will first describe a nomenclature for reflectance – reflectance can be expressed in various ways depending on the assumptions one is willing to make. Next, I describe how reflectance has been approximated using analytic formulas. The third section explains how reflectance of real-world surfaces has been measured in previous research. Next, I describe different representations for surface reflectance.

The second part of this chapter is devoted to dimensionality reduction techniques. I start with linear (subspace) methods that have been used for some time. Then, I move on to the recently developed non-linear (manifold) methods. These methods are an essential part of this thesis since I use them to derive my data-driven reflectance model.

2.1 Radiometry and Reflectance

Radiometry is the measurement of the electromagnetic radiation in the ultraviolet, the visible, and the infrared frequency spectrum. I will define the most important radiometric terms and then use them to describe the reflection of light from materials. I define reflectance as the ratio of the reflected light to the incident light at the surface. I consider the most general model of surface reflectance and explain the various simplifications and assumptions that lead to an isotropic BRDF model.

2.1.1 Radiometry

In this section I introduce the most important radiometric terminology and quantities. The descriptions follow the concepts defined in Nicodemus et al. [37], Jensen [18], and Marschner [32].

Radiation Flux

Radiation flux (or radiant flux, electromagnetic flux), usually denoted with Φ, is equivalent to power. It measures the time-rate flow of light energy:

$$\Phi = \frac{dQ}{dt}, \tag{2.1}$$

where Q denotes the energy of a collection of photons across all wavelengths and t denotes time. The unit of flux is the Watt [W].

Radiance

This is the most fundamental concept in radiometry. It is a physical quantity equivalent to the psychological concept of brightness observed by humans. It is usually denoted with the letter L and defined for all directions ω[1]. It measures electromagnetic flux $d\Phi$ travelling in the small range of directions through the solid angle element $d\omega$ and crossing an element of projected area dA (see Figure 2-1):

$$L(\omega) = \frac{d\Phi}{cos\theta dA d\omega}. \tag{2.2}$$

The unit of radiance is $\frac{Watt}{meter^2 \cdot steradian}\left[\frac{W}{m^2 \cdot sr}\right]$.

Irradiance

This quantity, denoted with the letter E, measures the differential incident flux falling onto a differential area of a surface. It is defined for all directions ω:

$$dE(\omega) = \frac{d\Phi}{dA}. \tag{2.3}$$

---

[1] Direction (ω) is usually denoted by two spherical coordinates elevation (θ) and azimuth (ϕ).

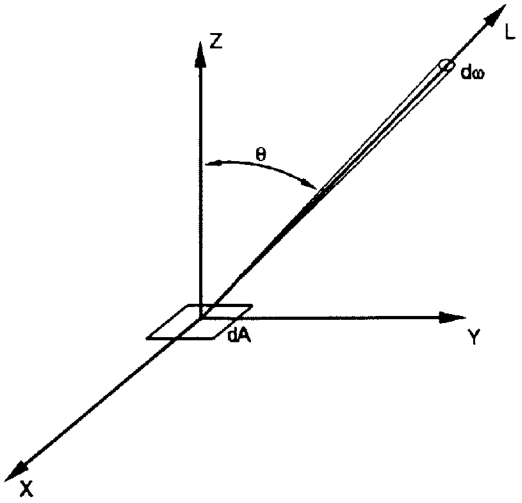

Figure 2-1: *Radiance (L) measures electromagnetic flux dΦ travelling in the small range of directions through the solid angle element dω and crossing an element of projected area dA*

Irradiance can be seen as a density of the incident flux falling onto a surface. It can be also obtained by integrating the radiance over the solid angle. The unit of irradiance is $\frac{Watt}{meter^2}\left[\frac{W}{m^2}\right]$.

2.1.2 Reflectance

Nicodemus et al. [37] defines reflection as the process by which electromagnetic flux incident on a stationary surface leaves the surface without a change in frequency. The reflectance is the fraction of the incident flux that is reflected. Phenomena like transmission, absorption, spectral effects, polarization, and fluorescence are not considered in this thesis. I consider the reflectance under the assumption of simple ray optics – the light travels only along the straight lines. Therefore, phenomena explained by wave optics (interference, diffraction) are also not considered.

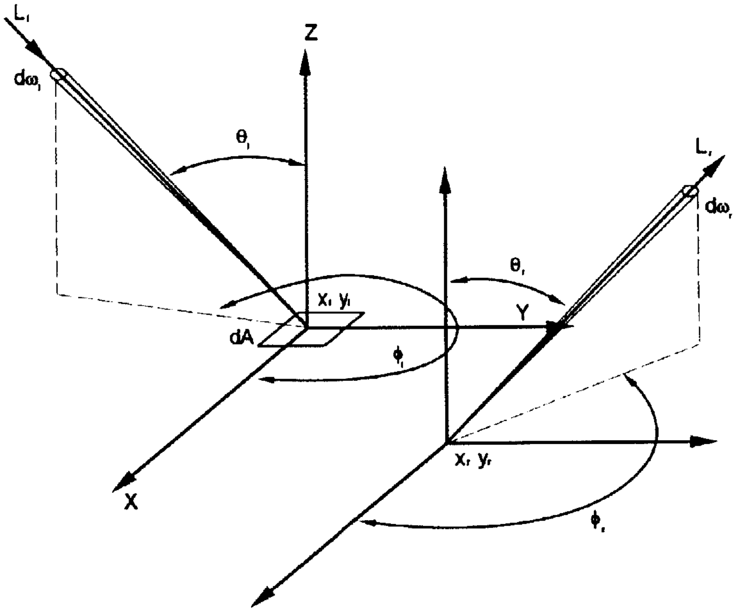

Figure 2-2: *BSSRDF, S, is defined as a ratio of reflected radiance* dL_r *in the direction* (θ_r, ϕ_r) *at a point* (x_r, y_r) *to incident flux* $d\Phi_i$ *coming from direction* (θ_i, ϕ_i) *at a point* (x_i, y_i).

BSSRDF

BSSRDF is a very general model of light transport. It considers all potential paths for the radiance seen emerging from a surface. Let's assume that the light reaches the material surface. Some fraction of it is absorbed and changed into heat. Some of the light is immediately reflected back in different directions. The rest scatters inside the material and some portion may exit the surface at different points. The material sample is not assumed to be homogenous; thus, each point on the surface might reflect the light in a different way. The function that describes how light reflects from a given surface is called the bidirectional scattering-surface reflectance distribution function (BSSRDF) [37], typically denoted with S. BSSRDF is a function of 8 variables. It is defined as a ratio of reflected radiance dL_r in the direction $(\theta_\mathrm{r}, \phi_\mathrm{r})$ at a point $(x_\mathrm{r}, y_\mathrm{r})$ to incident flux $d\Phi_\mathrm{i}$ coming from direction $(\theta_\mathrm{i}, \phi_\mathrm{i})$ at a point $(x_\mathrm{i}, y_\mathrm{i})$ (see Figure 2-2):

$$S(\theta_\mathrm{i}, \phi_\mathrm{i}, \theta_\mathrm{r}, \phi_\mathrm{r}, x_\mathrm{i}, y_\mathrm{i}, x_\mathrm{r}, y_\mathrm{r}) = \frac{dL_\mathrm{r}(\theta_\mathrm{r}, \phi_\mathrm{r}, x_\mathrm{r}, y_\mathrm{r})}{d\Phi_\mathrm{i}(\theta_\mathrm{i}, \phi_\mathrm{i}, x_\mathrm{i}, y_\mathrm{i})}. \tag{2.4}$$

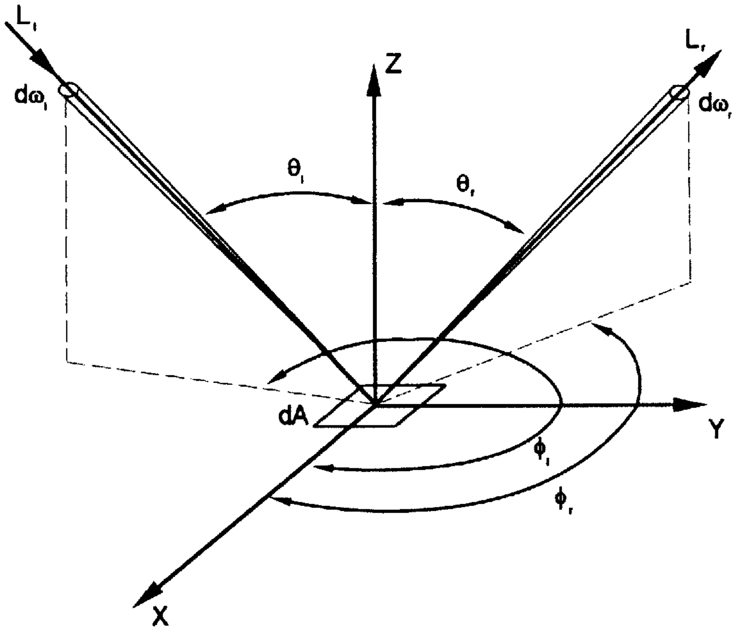

Figure 2-3: *BRDF, f_r, is defined as a ratio of incoming irradiance $dE_i(\theta_i, \phi_i)$ to the outgoing radiance $dL_r(\theta_r, \phi_r)$.*

The unit of BSSRDF is $\frac{1}{meter^2 \cdot steradian}\left[\frac{1}{m^2 \cdot sr}\right]$.

BRDF

One of the most useful models of reflectance is Bidirectional Reflectance Distribution Function (BRDF), denoted with f_r. It deals with the light that is immediately reflected when reaching the surface (the light that scatters inside the material and then leaves is not considered). This simplification of the BSSRDF takes advantage of the observation that, for most materials, the radiant flux incident at a point emerges from a point very near the point of incidence. This model of reflectance assumes that the surface is homogenous. BRDF is a function of four variables: two variables specify the incoming light direction, two other variables specify the outgoing light direction. It is defined as a ratio of incoming irradiance $dE_i(\theta_i, \phi_i)$ to the outgoing radiance $dL_r(\theta_r, \phi_r)$ (see Figure 2-3):

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{dL_r(\theta_r, \phi_r)}{dE_i(\theta_i, \phi_i)} = \frac{dL_r(\theta_r, \phi_r)}{L_i(\theta_i, \phi_i)\cos\theta_i d\omega_r}. \tag{2.5}$$

The unit of BRDF is $\frac{1}{steradian}\left[\frac{1}{sr}\right]$.

Isotropic BRDF

Isotropic BRDFs are an important subclass of BRDFs. The isotropic model is valid for materials for which rotations about the surface normal can be ignored. (The reflectance is the same when the material sample is rotated about the normal while the directions of the incoming irradiance and outgoing radiance are fixed.) In this case BRDF can be written as a function of only three variables. The variables ϕ_r and ϕ_i can be replaced by one variable $\phi_{diff} = (\phi_r - \phi_i)$. Therefore, the expression for isotropic BRDF can be expressed as follows:

$$f_r(\theta_i, \theta_r, \phi_{diff}) = \frac{dL_r(\theta_r, \phi_{diff})}{dE_i(\theta_i, \phi_{diff})} = \frac{dL_r(\theta_r, \phi_{diff})}{L_i(\theta_i, \phi_{diff}) \cos\theta_i d\omega_r}. \quad (2.6)$$

However, anisotropic surfaces – surfaces with preferred directions – cannot be modelled using isotropic BRDFs that are parameterized using only three angles. Anisotropic surfaces include brushed or burnished metals, hair, and fur. Kajija [21] gives as an example cloth that is composed of a weave of threads. Light which strikes threads along their length scatters differently than when it strikes them perpendicularly. Anisotropic surfaces have microgeometry with strongly oriented elements. The orientation of these elements causes the light to reflect in preferred directions relative to the local coordinate system expressed by the tangent, normal, and binormal. A distant observer does not see the microstructure, but instead only its effect on the reflected light. However, for the vast majority of the natural and man-made surfaces the microgeometry is randomly distributed and does not have any preferred direction.

2.1.3 Properties of BRDFs

BRDFs are not arbitrary functions. In order to obey the basic principles of physics they must satisfy certain constraints. I will describe the most important of these properties.

Non-negativity

All values of function f_r must be non-negative. In fact, they can be any value from the interval $[0,\infty)$. This is because both radiance and irradiance must be non-negative.

Energy Conservation

The energy conservation property can be stated in the following way:

$$\int_{\Omega} f_r(\theta_i, \phi_i, \theta_r, \phi_r)\, d\mu(\theta_r, \phi_r) \leq 1 \text{ for all } (\theta_i, \phi_i). \tag{2.7}$$

This property means that the amount of energy that is received by the surface element from some specific direction must be greater than the sum of the energy emitted by the surface element in all possible directions. This must be true for the energy received from all possible directions. The underlying assumption is that the surface element does not emit the energy by itself (e.g., the surface is not fluorescent).

Reciprocity

This property is also called Helmholtz's law of reciprocity. It states that the surface reflectance should be independent of the direction of the light flow – if the flow of light is reversed the value of the BRDF should be the same. This means that swapping the incoming and outgoing directions should yield the same value of f_r :

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r) = f_r(\theta_r, \phi_r, \theta_i, \phi_i). \tag{2.8}$$

Veach [54] has a detailed discussion of the reciprocity. He argues that reciprocity cannot be derived from the second law of thermodynamics or from the principle of time reversal invariance. However, reciprocity is widely observed and commonly assumed in reflection modelling.

2.2 Analytic Reflectance Models

Traditionally, in both the fields of computer graphics and computer vision, surface reflectance has been approximated with analytic reflectance models with varying degrees of complexity. These analytic reflectance models can be divided into two groups. (1) Phenomenological models – models that approximate the reflectance without analyzing underlying principles of physics. Phenomenological models are "ad hoc" empirical formulas that attempt to reproduce the typical reflectance properties seen in real surfaces. (2) Physically based models – these models try to make some simplifying assumptions about underlying physical properties of the surface. Physically based models usually model some specific phenomenon or specific types of materials (e.g., conductors).

2.2.1 Phenomenological Models

The phenomenological models initially used a simple cosine lobe [41] to approximate reflectance. They were modified to be physically plausible and more general [25]. Alternatively, a Gaussian instead of a cosine lobe has been used [55]. I will describe three phenomenological models that are commonly used in computer graphics.

Phong Model

The Phong model [41] is the most widespread reflectance model used in computer graphics. The original Phong model is not physically plausible – it does not satisfy energy conservation or reciprocity; however, there are simple modifications that ensure that these properties are met. The model is a sum of a diffuse component and a cosine-weighted specular component. It can be expressed as follows:

$$f_r(\hat{\mathbf{l}}, \hat{\mathbf{v}}) = k_d + k_s(\hat{\mathbf{v}} \cdot \hat{\mathbf{r}})^q / (\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}), \tag{2.9}$$

where $\hat{\mathbf{l}}$ is the unit vector towards the light; $\hat{\mathbf{n}}$ is the unit surface normal; $\hat{\mathbf{r}}$ is the reflected light direction; q is the specular reflection exponent; and k_d and k_s are the diffuse and specular coefficients. These parameters are specified separately for red, green, and blue colors, though the parameter q usually has the same value. Therefore, the Phong model uses a total of nine parameters.

Lafortune Model

The Lafortune model [25] can be seen as a generalized version of the Phong model. The model is able to express many phenomena that exist in the real world: non-Lambertian diffuse reflection (e.g., fading out of the diffuse component for grazing angles), Fresnel effect (e.g., increase in specularity at grazing angles), off-specular reflection, retro-reflection (scattering of the light back in the direction of the light source), and anisotropy. The physical plausibility of the model can also be easily enforced. The model can be expressed as follows:

$$f_r(\hat{\mathbf{l}},\hat{\mathbf{v}}) = \frac{k_d}{\pi} + k_s(\hat{\mathbf{l}}^{\mathrm{T}}\mathbf{M}\hat{\mathbf{v}})^q, \tag{2.10}$$

where **M** is a 3x3 symmetric matrix; $\hat{\mathbf{l}}$ is the unit vector towards the light; $\hat{\mathbf{v}}$ is the unit view vector; q is the lobe exponent; and k_d and k_s are the diffuse and specular coefficients. Next, the singular value decomposition is applied to matrix $\mathbf{M} = \mathbf{Q}^{\mathrm{T}}\mathbf{D}\mathbf{Q}$, where **D** is a diagonal matrix and **Q** is a transformation of the local coordinate system. The model can be rewritten as follows:

$$f_r(\hat{\mathbf{l}},\hat{\mathbf{v}}) = \frac{k_d}{\pi} + k_s(D_x l_x v_x + D_y l_y v_y + D_z l_z v_z)^q. \tag{2.11}$$

However, for isotropic BRDFs the values of parameters D_x and D_y are the same. Thus, a one lobe BRDF can be expressed using five parameters. Since each color component (red, green, blue) can have different parameter values, the Lafortune model requires at least 15 different parameters. (This is because representing a BRDF usually requires many lobes.)

Ward Model

The Ward model [55] is based on the elliptical Gaussian distribution (in contrast to the cosine-based distribution of Phong and Lafortune models). The model is very carefully designed to be physically plausible – it supports energy conservation and reciprocity. It is also relatively simple and can be evaluated efficiently. The parameters of the model have physical meaning and theoretically can be measured independently. The anisotropic Ward reflectance model is expressed as follows:

$$f_r(\theta_i,\phi_i,\theta_r,\phi_r) = \frac{k_d}{\pi} + k_s\frac{1}{\sqrt{\cos\theta_i\cos\theta_r}}\frac{\exp[-\tan^2\delta(\cos^2\phi/\alpha_x^2 + \sin^2\phi/\alpha_y^2)]}{4\pi\alpha_x\alpha_y}, \tag{2.12}$$

where k_d is the diffuse reflectance coefficient; k_s is the specular reflectance coefficient; δ is the angle between the half vector and the surface normal; ϕ is the azimuth angle of the half vector projected onto the surface plane; and α_x, α_y are the standard deviations of surface slope in the x, y directions, respectively. However, for isotropic BRDFs the values of parameters α_x and α_y are the same. Thus, the Ward model requires three parameters per wavelength – the total number of parameters is nine. Note that while the Ward model includes anisotropy, it does not model retro-reflection, or Fresnel effects, like the Lafortune model.

2.2.2 Physically Based Models

The value of physically accurate reflectance models has long been understood within the computer graphics community [4], but initially these models have been developed by applied physicists [51, 52]. Physical accuracy was an impetus behind the development of many subsequent computer graphics reflection models [7, 17]. I will describe in more detail three physically based reflectance models.

Cook-Torrance Model

The Cook-Torrance model [7] is a modification of earlier reflectance models [51, 50, 4]. The main assumption is that the surface is composed of tiny, perfectly reflective, smooth microfacets oriented at different directions. The facets are assumed to be V-shaped and their distribution is isotropic. The model takes into account the fact that the light might be blocked by other microfacets (shadowing). Similarly, it also considers the fact that the viewer does not see some of the microfacets since they are blocked by the other microfacets (masking effect). The model takes into account an average Fresnel term (polarization is not considered) when modelling the reflectance of individual microfacets. However, it does not allow for multiple light bounces between the microfacets. The orientation of the facets is assumed to have some distribution – Cook and Torrance use the Beckman distribution function. The reflectance of the surface depends on this distribution. The Cook-Torrance model can be expressed as follows:

$$f_{\mathrm{r}}(\theta_{\mathrm{i}}, \phi_{\mathrm{i}}, \theta_{\mathrm{r}}, \phi_{\mathrm{r}}) = \frac{k_d}{\pi} + \frac{k_s}{\pi} \frac{F\,D\,G}{\cos\theta_i \cos\theta_r}, \tag{2.13}$$

where k_d, k_s are diffuse and specular reflectance coefficients; F is the Fresnel factor; D is the microfacet distribution function; and G is the geometrical attenuation factor responsible for shadowing/masking. Thus, the Cook-Torrance model has five parameters for each color band – a total of 15 different parameters.

Oren-Nayar Model

Oren and Nayar developed a model [39] that approximates reflectance of rough diffuse surfaces. The surface is a collection of V-shaped microfacets. In contrast to the specular microfacets of the Cook-Torrance model, the microfacets are modelled as perfectly diffuse (Lambertian) surfaces. The cumulative reflectance of a collection of these facets is not Lambertian. Similarly to the Cook-Torrance model, the Oren-Nayar model takes into account shadowing and masking. However, it also considers interreflections between microfacets. The details of this model are not presented in this dissertation because they are too complex. This model uses four parameters for one color band.

He-Torrance-Sillion-Greenberg Model

This model [17] accounts for the phenomena that can be explained using both geometrical optics and wave optics (diffraction, interference). The model supports arbitrary polarization of incident light, but the simplifications for unpolarized light are also presented. In general, the reflectance is modelled as a sum of three components: specular, directional diffuse, and uniform diffuse. The specular component accounts for mirror-like reflection. It depends on the Fresnel reflectivity, roughness, and shadowing factors. The directional diffuse contribution of the reflectance function is the most complex term. It accounts for diffraction and interference effects. It depends on surface statistics (the effective roughness and the autocorrelation length). The uniform-diffuse contribution is a result of multiple microfacet reflections and subsurface reflections. It is expressed as a simple function of wavelength. The resulting isotropic reflectance model for unpolarized light is a function of four parameters. Each of the parameters has some physical meaning and (at least theoretically) can be measured separately. These parameters are: index of refraction, roughness, autocorrelation length, and uniform-diffuse factor. The model is too complicated to present. However, the number of intrinsic surface parameters is five for each color band – a total of 15 parameters.

While both phenomenological and physically based models have been widely used, there is a great deal of variance in the number of intrinsic surface parameters and the capabilities of each model. Another approach for modelling BRDFs for use in computer graphics simulations is to directly measure the reflectance of the desired material. The next sections describe how surface reflectance can be measured and then how it can be represented.

2.3 Reflectance Measurement

In the previous section I have discussed various analytic models that approximate reflectance of materials. One drawback of these models is the difficulty in finding proper parameter values for a desired material (e.g., wood, copper, etc.). One way to address this problem is to measure the reflectance of real materials. This section describes the research in reflectance measurement.

Traditionally, BRDFs have been measured using a gonio-reflectometer [38, 8]. The device consists of a light source and a detector. Both the light source and the detector can be placed at arbitrary directions with respect to the measured planar material sample. This is accomplished using mechanical elements (e.g., stage motors, robotic arms). The detector is usually a spectro-radiometer that measures the entire light spectrum reflected from the sample. The main drawback of using gonio-reflectometer is its inefficiency – only one BRDF value is measured at a time. Therefore, dense measurement of BRDF is impractical using this device.

Dana et al. [12] developed a system to measure spatially varying BRDFs, also called Bidirectional Texture Functions (BTFs). Using a digital camera, a robot arm, and a light source, they take approximately 200 reflectance measurements over varying incident and reflected angles for a planar material sample. The data for about 60 measured materials is available as the CUReT database [10]. With a uniform material sample this amounts to a relatively sparsely sampled BRDF. Such a sparsely sampled BRDF is not directly useful as a table-based BRDF function; thus, it is necessary to fit an analytic function in order to arrive at a useful model.

One of the first methods to speed up the measurement process is found in the pioneering work of Ward [55]. His measurement device (imaging gonio-reflectometer) consists of a hemispherical mirror and a CCD camera with a fisheye lens. The main advantage of his system is that the CCD camera can take multiple, simultaneous BRDF measurements. Each photosite of the imaging sensor contains a separate BRDF value (all these measurements have fixed incoming direction but different outgoing directions). Moving the light source and material over all incident angles enables the measurement of arbitrary BRDFs. Unfortunately, the device has some limitations: measurement of BRDF values near grazing angles is difficult; very specular BRDFs cannot be measured. More recently, Dana [11] proposed a device based on similar components and design. She uses a parabolic mirror and a low-cost Firewire camera with a regular lens.

Marschner et al.[34, 33] constructed another significant BRDF measurement system. The optical elements (mirrors) used by Ward and Dana to collect rays from different directions were replaced by a material sample with different surface normals. Each point with a different surface normal gave a different BRDF measurement. Their system used a spherical sample of homogenous material. A fixed camera took images of the sample under illumination from an orbiting light source. The system, although limited to only isotropic BRDF measurements, was both fast and robust. In particular, the system took unique advantage of reciprocity and multiple simultaneous measurements to achieve unprecedented leverage from each reflectance measurement. This offers a significant advantage. It filters measurement noise due to minute variations over the surface, errors due to spatial variations in photosite response within the image sensor, and variations in illumination intensity. Lu et al. [30] use a similar scanning device with cylindrical sample geometry to measure the anisotropic BRDF of velvet. Marschner et al. extended their method to surface geometry acquired with a laser range scanner, including human faces [34].

The measured samples of a BRDF are usually sparse. Therefore, they are typically fit to analytic BRDF models using various optimization techniques [55, 25, 12, 34]. Sato et al. [44] fit a spatially varying BRDF model to the relatively sparse image data of a rotating object with known geometry from laser range measurements. Lensch et al. [29] improve this approach by clustering sparsely sampled reflectance measurements, fitting a Lafortune BRDF model [25] to the data, and then computing basis BRDFs for material clusters using principal component analysis (PCA). Yu et al. [57] fit Ward's analytic BRDF model to real-world scenes that include global effects, such as indirect illumination.

One problem with fitting measured data to analytic reflectance models is that analytic reflectance functions are only an approximation of real reflectance, and the resulting analytic model is only an approximate fit to the measured BRDF values. Another approach is to treat reflectance as a "black box." The reflectance function can be stored in a tabulated form or it can be represented compactly using some basis functions. The next section describes various representation for reflectance functions.

2.4 Reflectance Representations

The inherent dimensionality of a BRDF, combined with the desire to sample it at high resolutions in order to model specular, incident, and retroreflection effects, leads to an unwieldy sampling and storage problem. Many researchers have addressed this specific problem by searching for a more appropriate basis for representing BRDFs.

As a result, many different BRDF representations have been developed. Westin et al. [56] proposed spherical harmonics to store simulated BRDF data. Lafortune et al. [25] approximates a BRDF with an arbitrary number of generalized cosine lobes. Schroeder et al. [45] use spherical wavelets to represent a slice of the BRDF with constant viewing direction. Lalonde and Fournier [27] use a wavelet decomposition and a wavelet coefficient tree to represent BRDFs. The major advantage of wavelets is they allow us to perform local analysis – that is, to analyze a localized area of a larger signal. Other efficient representations include Zernicke polynomials [23] and separable approximations obtained using singular value decomposition [22] or a purely positive matrix factorization [36]. Furthermore, recent image-based approaches to BRDF modelling [29] have demonstrated the power of using linear combinations of compact reflectance function basis sets for modelling spatially varying BDRFs.

2.5 Dimensionality Reduction Methods

In the second part of this chapter I examine dimensionality reduction methods. The problem of dimensionality reduction can be expressed as follows: Consider a process that generates high-dimensional data points. Each of these data points can be seen as D-dimensional vector ($x \in \mathbb{R}^{D}$). In many cases a given process will generate data that occupy only a small subset of the space $\mathbb{R}^{D}$ – all data points (that the process is capable of generating) reside on a low dimensional manifold embedded in the space $\mathbb{R}^{D}$. Dimensionality reduction methods discover this lower-dimensional manifold based on the set of the data points generated by the process. These methods are grouped into two categories: linear methods and non-linear methods.

2.5.1 Linear Methods

Linear methods assume that the high-dimensional data is embedded in a lower dimensional linear subspace (e.g., hyperplane). While this seems like a very limiting assumption, a large number of problems have been solved successfully using just linear dimensionality reduction tools. Linear methods have the advantage of fast and robust algorithms that compute the solution. They also need only a small number of data points to describe the subspace well. (In principle only $d + 1$ data points are required to describe d-dimensional linear subspace.)

Principal Component Analysis (PCA)

PCA or *Karhunen-Loéve transformation* is the predominant method for linear subspace estimation. The details of the method are described in Jolliffe [20]. I will outline the main points presented in Bishop [2]. The goal of PCA is to find a linear transformation that maps the D-dimensional original space onto d-dimensional space, where $d < D$, while preserving the most information about the data. For the sake of simplicity I assume that the data has zero mean. (If the mean of the data points is not zero, then it needs to be computed and subtracted from all data points.) Let $\mathbf{y}_i$ denote a vector representing a sample point in the D-dimensional space ($\mathbf{y}_i \in \mathbb{R}^{D}$) and matrix $\mathbf{Y}$ denote a set of all N points $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. These points can be expressed in a new orthonormal basis $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_D]$:

$$\mathbf{Y} = \mathbf{U}\mathbf{X}, \tag{2.14}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ denotes the coordinates of the data points in the new basis. Since $\mathbf{U}^\mathrm{T}\mathbf{U} = \mathbf{I}$, the matrix $\mathbf{X}$ can be computed from:

$$\mathbf{X} = \mathbf{U}^\mathrm{T}\mathbf{Y}. \tag{2.15}$$

Since the goal is to reduce the dimensionality required to represent the dataset, each vector $\mathbf{y}$ is approximated by $\tilde{\mathbf{y}}$ that uses only d basis vectors:

$$\tilde{\mathbf{y}}_i = \sum_{j=1}^{d} x_{ji}\, \mathbf{u}_j. \tag{2.16}$$

One measure of the information loss is the sum of the squares of the errors over the whole dataset E_M:

$$E_M = \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - \tilde{\mathbf{y}}_i\|^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=d+1}^{D} x_{ji}^2. \tag{2.17}$$

Using equation 2.15 the expression for E_M becomes:

$$E_M = \frac{1}{2}\sum_{j=d+1}^{D}\sum_{i=1}^{N}(\mathbf{u}_j^\mathrm{T}\mathbf{y}_i)^2 = \frac{1}{2}\sum_{j=d+1}^{D}\sum_{i=1}^{N}\mathbf{u}_j^\mathrm{T}\mathbf{y}_i\mathbf{y}_i^\mathrm{T}\mathbf{u}_j = \frac{1}{2}\sum_{j=d+1}^{D}\mathbf{u}_j^\mathrm{T}\Sigma\mathbf{u}_j \tag{2.18}$$

where Σ is the covariance matrix of all the points in the dataset:

$$\Sigma = \sum_{i=1}^{N}\mathbf{y}_i\mathbf{y}_i^\mathrm{T}. \tag{2.19}$$

It can be shown that E_M is minimized when vectors $\mathbf{u}_j$ correspond to the eigenvalue decomposition of the covariance matrix Σ:

$$\mathbf{U}\Lambda = \Sigma\mathbf{U} \tag{2.20}$$

$$\Lambda = \mathbf{U}^\mathrm{T}\Sigma\mathbf{U}, \tag{2.21}$$

Λ is a diagonal matrix of decreasing eigenvalues λ_j and $\mathbf{U}$ is an orthonormal matrix of corresponding eigenvectors. The minimum value of E_M is obtained when the vectors $\mathbf{u}_j$ in equation 2.22 correspond to the $D - d$ lowest eigenvalues:

$$E_M = \frac{1}{2}\sum_{j=d+1}^{D}\lambda_j. \tag{2.22}$$

This means that vectors $\mathbf{u}_j$ used in equation 2.16 need to be eigenvectors corresponding to the largest eignevalues.

However, when D (dimensionality of the original space) is large it is difficult or impossible to compute the covariance matrix Σ and its eigenvalue decomposition. I will show an alternative computation method using matrix $\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$. First, define matrix $\mathbf{B}$ and its eigenvalue decomposition:

$$\mathbf{B} = \mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \mathbf{V}\Lambda_{\mathbf{B}}\mathbf{V}^{\mathrm{T}}, \tag{2.23}$$

where $\mathbf{V}$ is the orthonormal matrix of eigenvectors and $\Lambda_{\mathbf{B}}$ is the diagonal matrix of decreasing eigenvalues. Next, left multiply both sides of equation 2.23 by $\mathbf{Y}$ and right multiply by $\mathbf{V}$:

$$\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{V} = \mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}\mathbf{V}^{\mathrm{T}}\mathbf{V} \tag{2.24}$$

$$\Sigma(\mathbf{Y}\mathbf{V}) = (\mathbf{Y}\mathbf{V})\Lambda_{\mathbf{B}}. \tag{2.25}$$

Next, right multiply both sides by $\Lambda_{\mathbf{B}}^{-\frac{1}{2}}$:

$$\Sigma(\mathbf{Y}\mathbf{V})\Lambda_{\mathbf{B}}^{-\frac{1}{2}} = (\mathbf{Y}\mathbf{V})\Lambda_{\mathbf{B}}\Lambda_{\mathbf{B}}^{-\frac{1}{2}} \tag{2.26}$$

$$\Sigma(\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}}) = (\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}})\Lambda_{\mathbf{B}}. \tag{2.27}$$

Note that matrix $\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}}$ is orthonormal:

$$(\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}})^{\mathrm{T}}(\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}}) = \Lambda_{\mathbf{B}}^{-\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}} = \Lambda_{\mathbf{B}}^{-\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\mathbf{V}\Lambda_{\mathbf{B}}\mathbf{V}^{\mathrm{T}}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}} = \mathbf{I}. \tag{2.28}$$

Comparing to equation 2.20 it must be that

$$\mathbf{U} = \mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}} \tag{2.29}$$

and

$$\Lambda = \Lambda_{\mathbf{B}}. \tag{2.30}$$

Furthermore, the coordinates $\mathbf{X}$ can be also computed more efficiently:

$$\mathbf{X} = \mathbf{U}^{\mathrm{T}}\mathbf{Y} = (\mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}})^{\mathrm{T}}\mathbf{Y} = \Lambda_{\mathbf{B}}^{-\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \Lambda_{\mathbf{B}}^{-\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\mathbf{V}\Lambda_{\mathbf{B}}\mathbf{V}^{\mathrm{T}} = \Lambda_{\mathbf{B}}^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}. \tag{2.31}$$

Classical (Metric) Multidimensional Scaling (MDS)

Multidimensional Scaling finds the linear subspace that best preserves the distances between all pairs of points. In the context of MDS, the distance between points is also called their dissimilarity. The input to the MDS algorithm is not the coordinates of the points in the high-dimensional space as in PCA. MDS instead takes as input a matrix **A** such that element $\mathbf{a}_{ij}$ of the matrix denotes a distance between point i and j in the dataset. To be metric, the distance measure must be non-degenerate ($\mathbf{a}_{ij} = 0$ iff $i = j$) and must preserve triangle inequality ($\mathbf{a}_{ij} + \mathbf{a}_{jk} \geq \mathbf{a}_{ik}$). In general, the distance measure can be user-defined; however, if the distances are Euclidean then MDS is equivalent to PCA. The first step in a MDS algorithm is to construct the inner-product matrix:

$$\mathbf{B} = -\frac{1}{2}\mathbf{HDH}, \tag{2.32}$$

where **D** is the matrix of squared distances between points and **H** is the centering matrix expressed as:

$$\mathbf{H} = \mathbf{I} - \mathbf{1}/N. \tag{2.33}$$

Next, the eigenvalue decomposition can be performed for the matrix **B**:

$$\mathbf{B} = \mathbf{V}\Lambda\mathbf{V}^{\mathrm{T}}, \tag{2.34}$$

where **V** is a matrix of eigenvectors and Λ is a diagonal matrix of decreasing eigenvalues. The coordinates of the points in the d-dimensional Euclidean space are given as columns of the matrix:

$$\mathbf{X} = \Lambda^{\frac{1}{2}}\mathbf{V}_{\mathbf{d}}^{\mathrm{T}}, \tag{2.35}$$

where matrix $\mathbf{V}_{\mathbf{d}}$ is a matrix composed of the eigenvectors corresponding to the d largest eigenvalues. Note that if d positive eigenvalues do not exist, then it is not possible to find a d-dimensional subspace for the points in the dataset.

Non-Negative Matrix Factorization (NMF)

The algorithm for NMF as described in Lee and Seung [28] works only on non-negative data. Its optimality constraints for dimensionality reduction are different from those of

PCA. (The components in PCA must be orhonormal while in NFM they need to be non-negative.) The input to the algorithm is a dataset of N examples where each example is a point in an D-dimensional space. The main assumption is that all coordinates of the examples are non-negative numbers. This assumption is valid for data produced by many physical processes (e.g., light is a positive process). All data can be concatenated to produce an $D \times N$ matrix $\mathbf{Y}$. The goal of NMF is to factor matrix $\mathbf{Y}$ into a product of two matrices as follows:

$$\mathbf{Y} \approx \mathbf{UX} \tag{2.36}$$

or each element of matrix $\mathbf{Y}$ is:

$$y_{ij} \approx (\mathbf{UX})_{ij} = \sum_{a=1}^{d} u_{ia} x_{aj}. \tag{2.37}$$

Columns of $\mathbf{U}$ are called basis vectors, and matrix $\mathbf{X}$ expresses the original examples in terms of these basis vectors. The main distinction from the PCA is the non-negativity of both matrices $\mathbf{U}$ and $\mathbf{X}$. Since $d < D$ the dataset is approximated using a lower dimensional non-negative basis. The main task of NMF is to find the optimal non-negative basis. In order to achieve this goal, the NMF algorithm maximizes the objective function:

$$F = \sum_{i=1}^{D} \sum_{j=1}^{N} [y_{ij} \log(\mathbf{UX})_{ij} - (\mathbf{UX})_{ij}]. \tag{2.38}$$

Unfortunately there is no closed-form solution for matrices $\mathbf{U}$ and $\mathbf{X}$. Thus, the algorithm is iterative. It updates matrices $\mathbf{U}$ and $\mathbf{X}$ in a way that guarantees convergence on the local maximum.

The use of NMF in the context of BRDFs is rational since BRDFs are descriptions of a natural and non-negative process. Furthermore, the basis vectors obtained using NMF are valid BRDFs. They could, in principle, correspond to some prototypical BRDFs (e.g., diffuse BRDF, specular BRDF, etc.).

2.5.2 Non-linear Methods

The input to the non-linear dimensionality reduction algorithms (called Non-linear Dimensionality Reducers – NLDRs) is a set of N sample points $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, where each point

$\mathbf{y}_i \in \mathbb{R}^{\mathrm{D}}$ (i.e., the same input as with linear dimensionality reduction). The assumption is that these points are samples coming from some manifold that has dimensionality d that is lower than D. The manifold learning techniques attempt to discover a mapping from the original space (also called sample or ambient space) to the embedding space (also called target space) $G(\mathbf{Y}) \rightarrow \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, where $\mathbf{x}_i \in \mathbb{R}^{\mathrm{d}}$. The mapping should preserve the local relationship between the sample points. The long distance relationships are presumed to be corrupted by the curvature of the manifold in the ambient space. The mapping should generate the manifold without folds. (Parallel lines in the ambient space in general will not be parallel in the embedding space; however, they should map to smooth, continuous, and non-intersecting curves.) The mapping can be successfully computed using first generation NLDRs such as nonmetric MDS [24], IsoMap [49], and LLE [42] for a large class of different non-linear manifolds. The second generation algorithms [48, 5] also compute the reverse mapping $G^{-1}(\mathbf{X}) \rightarrow \mathbf{Y}$. The mapping projects the embedding space back to the ambient space.

IsoMap Method

I will give a brief overview of the IsoMap algorithm as presented in Tenenbaum et al. [49]. The algorithm is one of the first non-linear dimensionality reducers. The input is a collection of high-dimensional data points. First, the authors note that for nearby points the input space distance is a good approximation of the distance on the manifold (geodesic distance). However, for far away points this approximation is not adequate. In their approach, the authors try to establish all geodesic distances between far away points by solving a shortest path graph problem using the distances between nearby points. This assumes that the graph has a single connected component. Next, they require a dimensionality reduction method that preserves all distances. Thus, their method uses classical MDS, which is designed to preserve the structure (distances) of the original data.

The input to the IsoMap algorithm is the $N \times N$ distance matrix D_y, where N is the number of points in the dataset. Each element (i, j) of the matrix is equal to the distance between the data points i and j, denoted as $d_y(i, j)$. The distance metric is Euclidean but alternatively can be user-defined. The other inputs to the algorithm are: K - minimum neighborhood size and ε - maximum neighborhood distance. The IsoMap algorithm proceeds in three steps that are outlined below:

- **Neighborhood Graph Construction:** The nodes of the neighborhood graph G correspond to the N points in the dataset. The edges and their weights are set in two ways: the edge with the weight $d_v(i,j)$ is set if and only if (1) the distance between points i and j is less than ε; or (2) the point j is one of the K nearest neighbors of point i.

- **All-pairs Shortest Path Computation:** Next, the geodesic distance matrix D_G is initialized based on the neighborhood graph. The element (i,j) of the matrix is set to the weight of the edge (i,j) in graph G. If the edge does not exist the element is set to ∞. Next, all-pairs shortest paths are computed [15] between all data points by updating matrix D_G. Thus, at the end the matrix D_G stores the shortest distances between all points in the dataset. Note that it is possible that the matrix D_G still contains some ∞ values. This means that some nodes are not accessible from others. In this case, (1) the disconnected node sets are removed from further analysis; (2) the analysis is conducted on separate disconnected node sets; or (3) the values of K and/or ε are increased.

- **Embedding Space Construction:** In the final step of the algorithm, MDS is applied to the matrix D_G in order to compute the d-dimensional Euclidean embedding space. First, the distance matrix D_G is converted to inner-product matrix $\mathbf{B}$ (equation 2.32). Then, the eigenvalue decomposition of matrix $\mathbf{B}$ is performed. The coordinates of the dataset's d-dimensional embedding is obtained by using the d-largest eigenvectors and eigenvalues (equation 2.35) .

It is important to note that the IsoMap method provides a mapping only from the original (ambient) space to the embedding space. The method to compute the reverse mapping (from embedding space back to the ambient space) is not clearly defined.

Locally Linear Embeddings (LLEs)

The algorithm for computing locally linear embeddings was originally presented by Roweis and Saul [42]. I will highlight the most important points of the algorithm. This method also tries to preserve the local structure of the manifold but in a different way than the IsoMap algorithm. Each point in the dataset is expressed as a linear combination of its closest neighbors. The mapping of points to the embedding space tries to preserve the coefficients of the linear combinations. The method for computing locally linear embbedings has three steps:

- **Selecting Neighbors:** First, each point in the dataset has its K- or ε-based nearest neighbors assigned. This step is similar to the first step of the Isomap method.
- **Reconstruction with Linear Weights:** Next, each point in the high-dimensional space is reconstructed as a linear combination of its nearest neighbors. Let $\mathbf{y}_i$ denote point i in the dataset and W_{ij} denote reconstruction weights for point i of point j. Then,

  $$\mathbf{y}_i = \sum_{j \in \text{ neighbors of i}}^{K} W_{ij}\mathbf{y}_j \tag{2.39}$$

  and

  $$\sum_{j \in \text{ neighbors of i}}^{K} W_{ij} = 1. \tag{2.40}$$

  The weights for each point are computed by solving a constrained least squares problem. The reconstruction weights characterize the geometric structure of the manifold.
- **Mapping to Embedding Space:** In the last step of the algorithm, a low dimensional embedding space is computed and all high dimensional point coordinates $\mathbf{y}_i$ are mapped to low dimensional coordinates $\mathbf{x}_i$. The low-dimensional coordinates of point i are set to preserve the the reconstruction weights. This is achieved by minimizing the cost function:

  $$\Phi(\mathbf{X}) = \sum_{i=1}^{N} |\mathbf{x}_i - \sum_{j \in \text{ neighbors of i}}^{K} W_{ij}\mathbf{x}_j|^2. \tag{2.41}$$

The solution is obtained using the eigenvalue decomposition of matrix **M**, where each element can be expressed as follows:

$$m_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj} \tag{2.42}$$

and

$$\delta_{ij} = \begin{cases} 1 & \text{i=j} \\ 0 & \text{i} \neq \text{j}. \end{cases} \tag{2.43}$$

The coordinates of the points in the original dataset expressed in the d-dimensional embedding space are obtained from the d eigenvectors corresponding to the smallest d eigenvalues[2].

Note that this method does not require solving an expensive all-pairs shortest path problem; thus, the method should be faster than the IsoMap algorithm. Furthermore, the initial version of the method does not provide the mapping from embedding space back to the ambient space.

Automatic Alignment [48] combines LLE with a set of pre-estimated local dimensionality reducers – each of which is presumed to be fitted to a relatively flat subset of the manifold – and solves for a mixture of these projections that globally flattens the data while minimizing barycentric distortion in each point neighborhood.

Manifold Charting

Manifold charting was introduced by Brand [5]. This is the method I use to perform the analysis; therefore, I describe it in more detail. Charting solves for a kernel-based mixture of projections that minimizes Euclidean distortion of local neighborhoods. The algorithm can be divided into two major steps. (1) Charting the data – the first step computes a set of charts (a chart is a locally linear neighborhood). The charts cover the whole dataset. Neighboring charts span maximally similar subspaces. (2) Connecting the charts – the second step computes a minimal-distortion merger of all charts to obtain the mappings $G(\mathbf{Y}) \rightarrow \mathbf{X}$ and inverse mapping $G^{-1}(\mathbf{X}) \rightarrow \mathbf{Y}$. Figure 2-4 gives the main geometric intuition behind charting.

---

[2] Note that the smallest eigenvector (the eigenvalue 0) corresponds to a global translation.

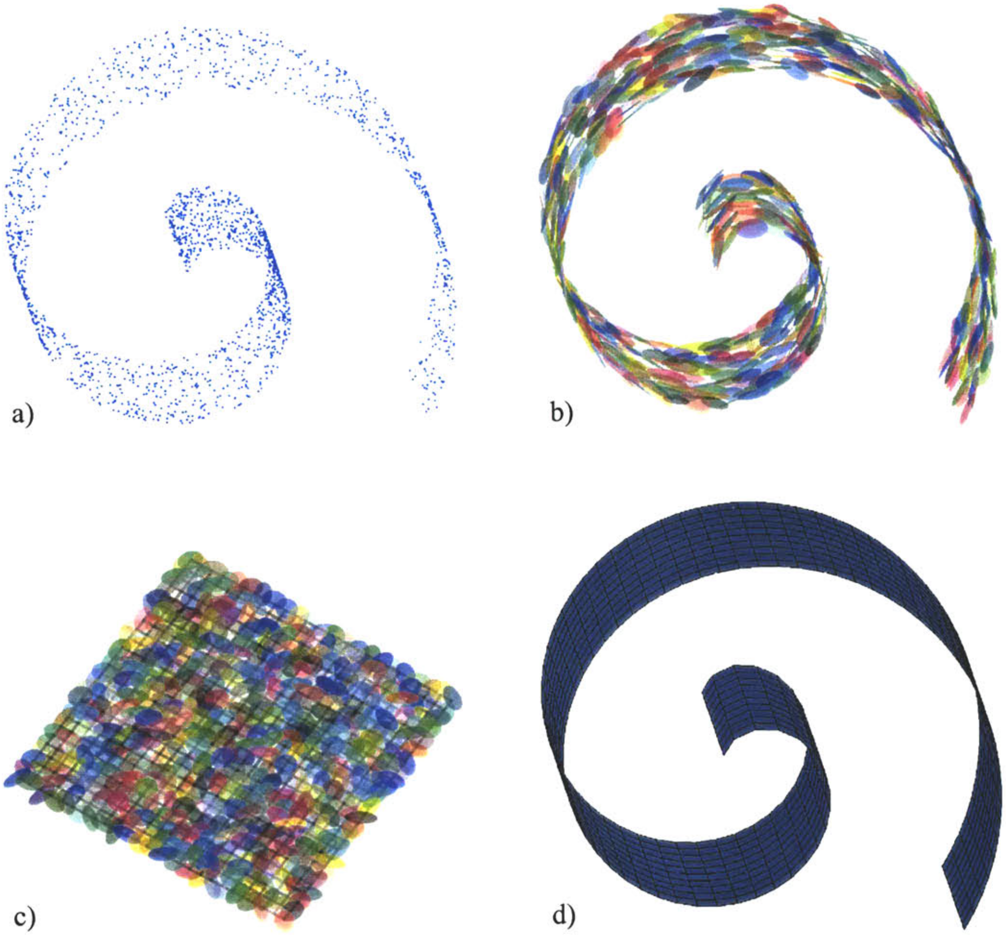

Figure 2-4: *A simple charting example. (a) Points are sampled from an unknown manifold. (b) In the charting step linear subspaces and soft partitioning of the data points is computed. (c) The next step is to compute a global coordinate system that introduces minimal local distortion to the neighborhoods. This effectively flattens the manifold onto a low-dimensional subspace. (d) The manifold can be reconstructed by computing a back-projection function.*

Charting the Data: This step computes both soft partitioning of the data points into locally linear subspaces and these linear subspaces. Charting assumes that the data can be modelled by a mixture of Gaussians. It fits a Gaussian Mixture Model (GMM) to the data points to maximize the likelihood function:

$$p(\mathbf{y}_i|\mu,\Sigma) = \sum_j \mathcal{N}(\mathbf{y}_i;\mu_j,\Sigma_j)p_j, \tag{2.44}$$

where μ_j and Σ_j denote mean and covariance of each Gaussian; p_j denotes mixing proportions of each Gaussian. This ensures the minimal loss of local variance in the data. However, in order to minimize the information loss, the locally linear subspaces that are adjoining each other should span nearly the same subspace (but their dominant axes do not have to be aligned). This is equivalent to maximizing the cross-entropy between the neighboring Gaussians. Charting maximizes consistency between neighboring linear subspaces by forming the prior:

$$p(\mu,\Sigma) = \exp[-\sum_{i\neq j} m_i(\mu_j)D(\mathcal{N}_i||\mathcal{N}_j)], \tag{2.45}$$

where $D(\mathcal{N}_i||\mathcal{N}_j)$ is a cross-entropy between two Gaussians, and $m_i(\mu_j)$ is a measure of co-locality. Under these assumptions the posterior probability is expressed as follows:

$$p(\mu,\Sigma|\mathbf{Y}) = \sum_i p(\mathbf{y}_i|\mu,\Sigma)p(\mu,\Sigma). \tag{2.46}$$

If the mixture has one Gaussian for each data point then $\mu_i = \mathbf{y}_i$ and all Gaussians are equally probable ($p_i = 1/N$). Furthermore, the measure of co-locality can be expressed as:

$$m_i(\mu_j) = \mathcal{N}(\mu_j;\mu_i,\sigma^2), \tag{2.47}$$

where σ specifies the expected size of a neighborhood on the manifold. (A reasonable value for σ is half the average distance between each point and its closest neighbor.) Under these assumptions the MAP estimates for covariances of individual Gaussians are:

$$\Sigma_i = \frac{(\Sigma_j m_i(\mu_j)((\mathbf{y}_j-\mu_i)(\mathbf{y}_j-\mu_i)^{\mathrm{T}} + (\mu_j-\mu_i)(\mu_j-\mu_i)^{\mathrm{T}} + \Sigma_j))}{\Sigma_j m_i(\mu_j)}. \tag{2.48}$$

The closed form solution for these covariances is computed by solving a linear system of equations.

Connecting the Charts: Once the charts have been computed the next step is to compute a global low-dimensional coordinate system that introduces the minimal local distortion to the neighborhoods. First, the eigenvalue decomposition is performed for the covariance matrix for each Gaussian:

$$\Sigma_i = \mathbf{V}_k \Lambda_k \mathbf{V}_k^{\mathrm{T}}, \tag{2.49}$$

where $\mathbf{V}_k$ is the eigenvector matrix and Λ_k is the diagonal matrix with decreasing eigenvalues. Let $\mathbf{W}_k$ be an operator that projects points in the original space onto the d-dimensional space of the k^{th} chart:

$$\mathbf{W} = [\mathbf{I}_d, 0]\mathbf{V}_k^{\mathrm{T}}. \tag{2.50}$$

Then the coordinates of the point $\mathbf{y}_i$ in the coordinate frame of the k^{th} chart can be computed as follows:

$$\mathbf{u}_{ki} = \mathbf{W}_k(\mathbf{y}_i - \mu_k). \tag{2.51}$$

A matrix $\mathbf{U}_k$ is formed by concatenating local coordinates for all points in the dataset. For each chart there is an affine transform $\mathbf{G}_k \in \mathbb{R}^{(d+1)\times d}$ that maps the local coordinate system to the global coordinate system. Since each point in the original space has a soft label assignment for each chart, the final mapping from the ambient to embedding space is a simple mixture of affine projections, weighted by the probability that a point "belongs" to each chart:

$$\widehat{\mathbf{x}_i|\mathbf{y}_i} = \sum_j \mathbf{G}_j \begin{bmatrix} \mathbf{u}_{ji} \\ 1 \end{bmatrix} p_{j|\mathbf{y}}(\mathbf{y}_i), \tag{2.52}$$

where $p_{k|\mathbf{y}}(\mathbf{y}) = p_k \mathcal{N}(\mathbf{y}; \mu_k, \Sigma_k)$ and $\sum_k p_{k|\mathbf{y}}(\mathbf{y}) = 1$. Thus, the main task of this step is to compute all affine transforms $\mathbf{G}_k$ that connect the charts. The data points that have non-zero probability in two different charts should be mapped to the same point in the global coordinate system. Therefore, the affine transforms should be computed such that the sum of squared differences in the points' position is minimized:

$$\mathbf{G} = [\mathbf{G}_1, \ldots, \mathbf{G}_K] = \arg\min_{\mathbf{G}_k, \mathbf{G}_j} = \sum_i p_{k|\mathbf{y}_i}(\mathbf{y}_i) p_{j|\mathbf{y}_i}(\mathbf{y}_i) \left\| \mathbf{G}_k \begin{bmatrix} \mathbf{u}_{ki} \\ 1 \end{bmatrix} - \mathbf{G}_j \begin{bmatrix} \mathbf{u}_{ji} \\ 1 \end{bmatrix} \right\|^2 . \tag{2.53}$$

Brand [5] shows two different solutions for this weighted least squares problem; they both reduce to computing the eigenvalue decomposition of a matrix. Finally, the inverse mapping gives a smoothly curving low-dimensional surface in the ambient space, effectively reconstructing the original manifold. The back-projection operator (the posterior-mean back-projection) can be defined as follows:

$$\widehat{\mathbf{y}|\mathbf{x}} = \sum_k p_{k|\mathbf{x}}(\mathbf{x}) \left(\mu_k + \mathbf{W}_k^{\mathrm{T}} \left(\mathbf{G}_k \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \right)^+ \left(\mathbf{x} - \mathbf{G}_k \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \right) \right), \tag{2.54}$$

where $(\cdot)^+$ denotes pseudo-inverse and $p_{k|\mathbf{x}}$ can be obtained from:

$$p_{k|\mathbf{x}}(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}; \mathbf{G}_k \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \mathbf{G}_k \begin{bmatrix} [\mathbf{I}_d, \mathbf{0}]\Lambda_k[\mathbf{I}_d, \mathbf{0}]^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \mathbf{G}_k^{\mathrm{T}}\right). \tag{2.55}$$

It is worth to mention that this back-projection operator is not 1-1.

2.6 Summary

In this chapter I have outlined the major areas of previous work. First, I introduced basic concepts in radiometry and reflectance definitions. Then, I described analytic reflectance models, measurement, and representations. The second central area of the previous work chapter covers unsupervised learning methods that are used for dimensionality reduction. I reviewed major linear and non-linear (manifold) methods. In the next chapter I will describe a measurement system used to acquire densely sampled isotropic BRDFs for a large collection of materials.

Chapter 3

Measurement and Data Representation

This chapter describes the construction of my BRDF measurement system. I discuss the whole acquisition process. This includes geometric calibration, radiometric measurements, and computation of actual BRDF samples. Next, I describe parameterization and representation of the acquired BRDF data – reflectance of each material is represented as a data structure composed of dense BRDF samples. In this way I measure and represent reflectance for a large collection of different materials.

3.1 Measurement System

In order to acquire a sufficient number of adequately sampled BRDFs, it was necessary to build a measurement device. My modelling approach placed two requirements on the acquired data: first, each BRDF should be sampled densely enough to be used directly as a table-based model, and second, the space of BRDFs should be sampled adequately so as to span the range of surface reflectances that I hope to generate. Accordingly, I have built a BRDF measurement device suitable for rapidly acquiring high-quality isotropic BRDFs for a wide range of different materials. My design was inspired by Marschner's image-based BRDF measurement device [33]. His device, like mine, requires a spherically homogenous sample of material.

A photograph of my measurement system is shown in Figure 3-1 and its schematic is shown in Figure 3-2. The system is placed in a completely isolated room, with all

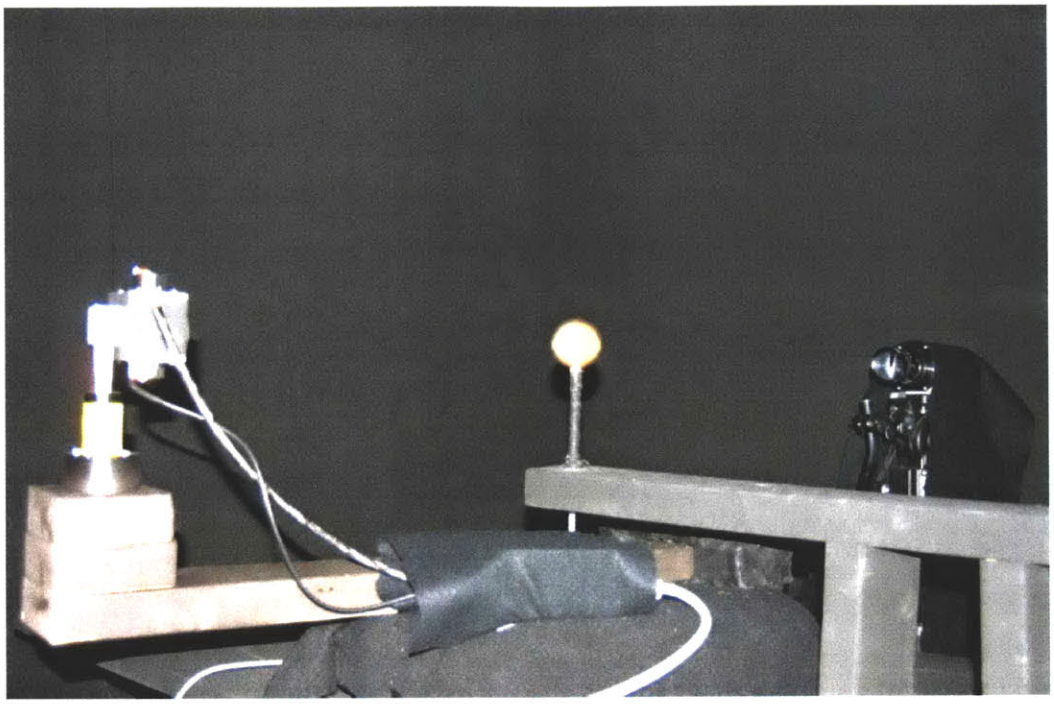

Figure 3-1: *A photograph of my high-speed BRDF measurement gantry.*

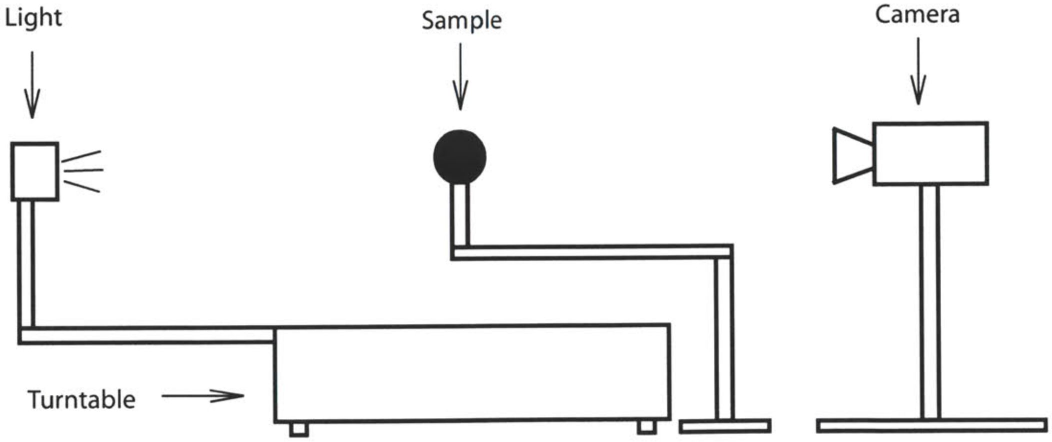

Figure 3-2: *A schematic of my high-speed BRDF measurement gantry.*

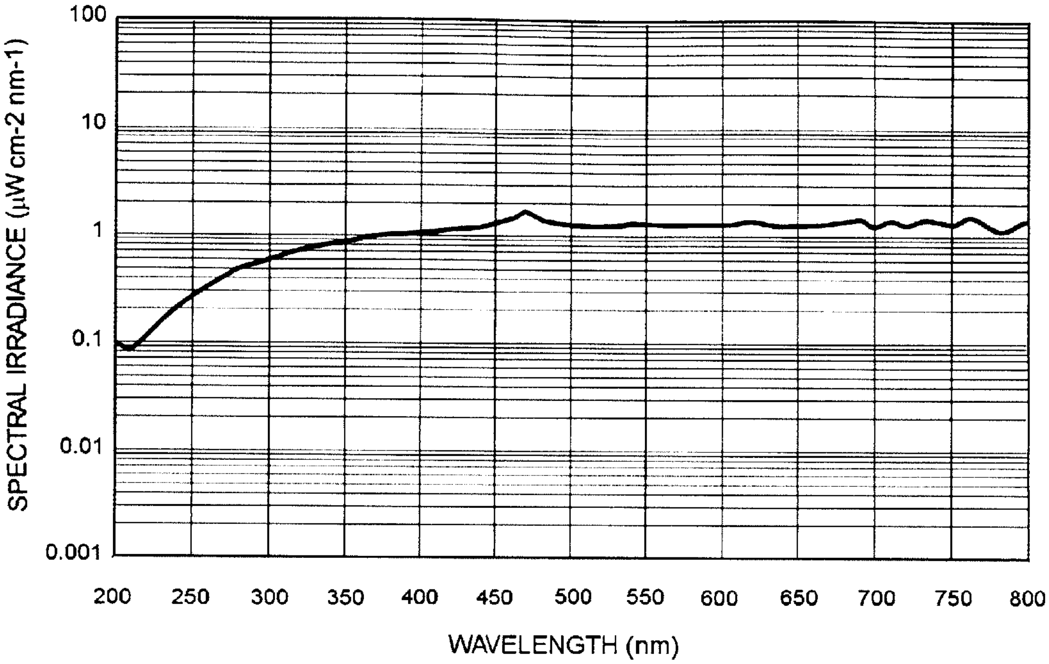

Figure 3-3: *Spectral Irradiance 50 cm from the light source.*

of the surfaces of the instrument covered in either black felt or black construction paper. In addition, all walls of the room are painted a matte black. It consists of the following components: a QImaging Retiga 1300 (a 10-bit, and a 1300 × 1030 resolution Firewire camera), a Kaidan MDT-19 (a precise computer-controlled turntable), and a Hamamatsu SQ Xenon lamp (a lamp with stable light emission output and a continuous and relatively constant radiation spectrum over the visible light range – Figure 3-3). The lamp is mounted on an arm to a turntable. The lamp orbits the measurement sample placed at the center of rotation; the camera and the sample are stationary. The light source moves in increments of approximately 0.5° from the point opposite the camera (the sample is in between the camera and the light source) to the point in front of the camera. I take a total of 330 high-dynamic range pictures to cover the required half circle. This process takes about four hours.

3.2 Geometric Calibration

Before proceeding with BRDF measurements, the relative positions of the camera, light source, and material sample must be found. In addition, the imaging parameters of the camera and the precise size of the material sample are determined. This process is called geometric calibration.

In order to geometrically calibrate the camera I need to determine its internal and external parameters. The internal parameters are computed using the technique developed by Zhang [58]. The external parameters are computed using about 30 correspondences between the 3D coordinates of the tip of a contact digitizer (FARO Arm) and its 2D position in the image. Thus I obtain the relationship between image pixels and the corresponding rays in 3D.

Next I determine the locations of the light source. The light source is assumed to move on a circle in 3D. Therefore, I have to determine the circle parameters and a reference light position on the circle. First, I move the light source and measure approximately 30 of its locations using a contact digitizer. Then, I determine the plane of the circle (this requires solving an overconstrained system of linear equations). Next, I project all the 3D locations onto this plane and compute the equation of the circle in 2D using a non-linear optimization [31]. I obtain the circle center and a vector from the circle center to the current position of the light source. Using this parameterization I can move the light source using the turntable and determine its precise location in 3D.

Computing the position of the material sample is relatively simple. First, I measure the diameter of the sample using calipers. Next, I measure the locations of about 30 points on the surface of the sample using a contact digitizer. Then, I use non-linear optimization [31] to find the center of the sample.

3.3 High-Dynamic Range Radiance Measurements

The dynamic range of the scene exceeds the 10-bit range of Retiga 1300. Therefore, I use multi-exposure photography to derive high-dynamic range images of the material sample.

For each high-dynamic range picture I take a total of 18 10-bit photographs with increasing exposure times. The exposure time increases from 40 microseconds to 20 seconds. I take advantage of the fact that CCD cameras have a linear response curve; thus, I do not need to recover the response curve of the camera [13].

For each pixel in the image I have 18 measurements. I discard all measurement values below 10 and above 1000. I then fit a line to the exposure time vs. measured radiance values for the remaining measurements. The slope of the line is used as the radiance estimate. The correlation of this line for all pixels in the image is higher than 0.998.

Some of the CCD imager sensor elements (photosites) have high dark current values. This means they generate a significant output even in the absence of any incoming visible radiation. These dark current levels are a function of the ambient room temperature, localized impurities, and processing variations in the CCD's construction. As a result, these elements do not produce valid measurements. In order to detect these elements I capture one image with 60 second exposure time and a closed lens cover (no external light gets to the imager). The elements that have values above 16 are considered to have excessive dark current values, and are not used in the BRDF measurement process.

3.4 BRDF Computation

Each acquired image of the material sample represents many simultaneous BRDF samples. Each imaged pixel of the sphere is treated as a separate BRDF measurement. This measurement is either red, green, or blue since the Retiga 1300 produces a Bayer pattern image. (I do not attempt to interpolate the other color components using nearby sensor elements.) In order to compute the specific BRDF value for a given image pixel I perform the following steps. First, I intersect the ray defined by the pixel with the sphere to determine point *P*. Then, I compute the normal at point *P* on the sphere, the vector and the distance to the light source, and the vector to the camera pixel. Next, I compute the irradiance at point *P* due to the light source (taking into account distance to the light source and foreshortening). Finally, I compute the BRDF value as the ratio of the high-dynamic range radiance to the irradiance.

3.5 Alternative BRDF Computation

The method of computing BRDF samples described in the previous section does not produce samples at exactly specified values of (θ_r, θ_i, ϕ_{diff}). An alternative way to compute BRDF samples would be to compute an inverse mapping: given fixed values (θ_r, θ_i, ϕ_{diff}) I would like to determine the image number in the sequence and the coordinates in this image that store the radiance measurement needed to compute the BRDF value.

In Figure 3-4, I show the geometric construction that allows me to compute this mapping efficiently. First, I assume that each image in the sequence is continuous – radiance can also be evaluated at non-integer locations. Note that the sphere center, position of the light source, and camera center of projection form a plane in 3D. I observe that all points that have the same value of θ_r lie on a circle that resides on the sphere and is centered on the line segment between the camera center of projection (COP) and the sphere center (see Figure 3-4). Similarly, all points with the same value of θ_i lie on a circle that resides on the sphere and is centered on the line segment between the light source and the sphere center. It follows that the points that have some fixed values of θ_r and θ_i lie on the intersections of these two circles. The two circles can have zero, one, or two intersection points. The exact solution can be determined by solving a quartic equation. Given a discrete intersection point I can compute the remaining parameter ϕ_{diff}. This means that although we can freely pick values of θ_r and θ_i, the value of ϕ_{diff} is discrete. This is caused by the discrete positions of the light source in the sequence of images. My mapping algorithm computes the the intersections of the desired circles for all light configurations and the corresponding ϕ_{diff} values. Then, the position for the closest to the desired ϕ_{diff} value is returned. This method of computing BRDF samples assumes that it is possible to obtain radiance values for any, possibly non-integer, point coordinates in the image. Therefore, these values need to be interpolated from the values at the discrete pixel locations.

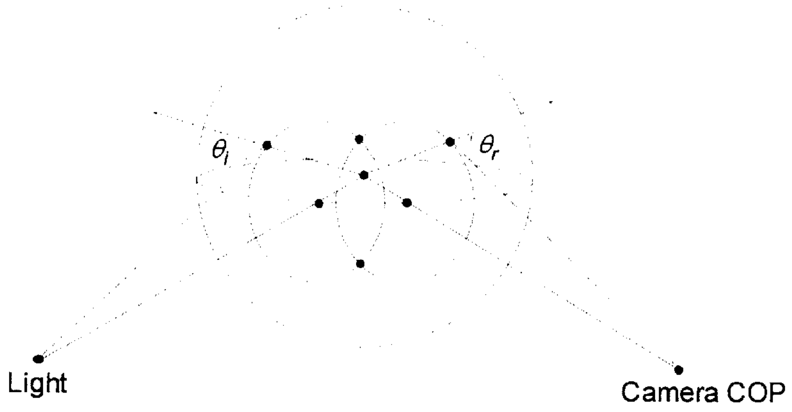

Figure 3-4: *Geometric construction used to compute mapping between BRDF coordinates and sampling rays.*

3.6 Data Representation

The specular peaks of highly reflective materials are particularly difficult to represent using the natural isotropic BRDF parameterization (θ_r, θ_i, ϕ_{diff}). Even when binning a BRDF at a dense grid (every 1 ° spacing for each dimension), it is not possible to reproduce original images (the specular highlight becomes an oval shape, oriented at different directions). In order to address these sampling problems, I used a different coordinate system, introduced by Rusinkiewicz [43] and illustrated in Figure 3-5. This coordinate frame is based on the angles with respect to the half-angle (half-vector between incoming and outgoing directions). The half-angle is a relevant reflection parameter because it defines an ideal surface normal for which a mirror surface (perfect reflector) would reflect all of the incoming light in the specified outgoing direction. This coordinate frame allows me to vary the sampling density near the specular highlight. Specifically, I vary θ_h (angle between the actual surface normal and the half-vector), assigning smaller bins for values near specular reflection and larger bins for angles far away from the specular reflection[1].

I still subdivide θ_h, θ_d into 90 bins and ϕ_d into 360 bins. This results in a total of 90 x 90 x 360 = 2,916,000 bins for each color component. I half this number to 1,458,000 by

---

[1] I use $\theta_h = 0.2 * 1.07^{bin}$.

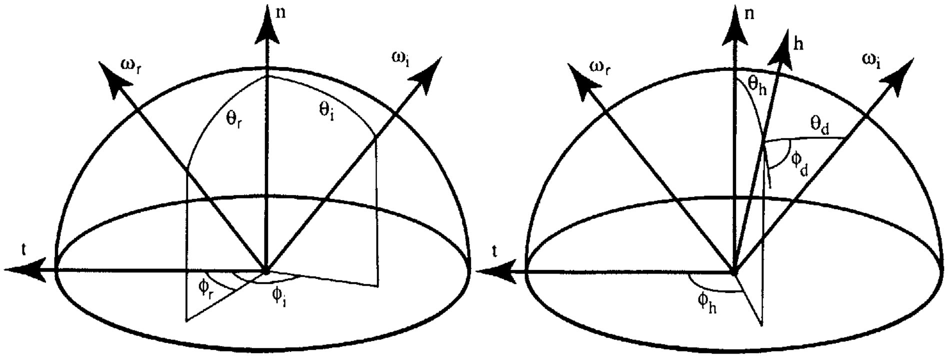

Figure 3-5: *The standard coordinate frame is shown on the left. The changed coordinate system is shown on the right.*

enforcing the reciprocity constraint:

$$f(\theta_h, \theta_d, \phi_d) = f(\theta_h, \theta_d, \phi_d + \pi) \tag{3.1}$$

With this constraint I need only to subdivide ϕ_d into 180 bins.

My measurement process gives me typically 20-80 million BRDF samples for each material. I reduce the noise in the measurements by removing the outliers in each bin (lowest and highest 25% of the values), and I average the remaining measurements. This statistical smoothing is intended to remove systematic noise as well as compensate for small variations in material properties over the sample. As a final validation I render a synthesized version of our sample sphere and compare it to the corresponding acquired high-dynamic range image. I conduct this inspection for all input light configurations. Pictures for a typical acquired material are in shown in Figure 3-6. The rendered images reproduce the input images very well. I have used my device to acquire BRDF measurements of more than 130 different materials, including metals, plastics, painted surfaces, and cloth. Figure 3-7 depicts some of the materials that were sampled. Upon analyzing the data I removed some materials that exhibited significant subsurface scattering, anisotropy, or nonhomogeneity (using the isotropic BRDF model it is not possible to reproduce input images for these materials).

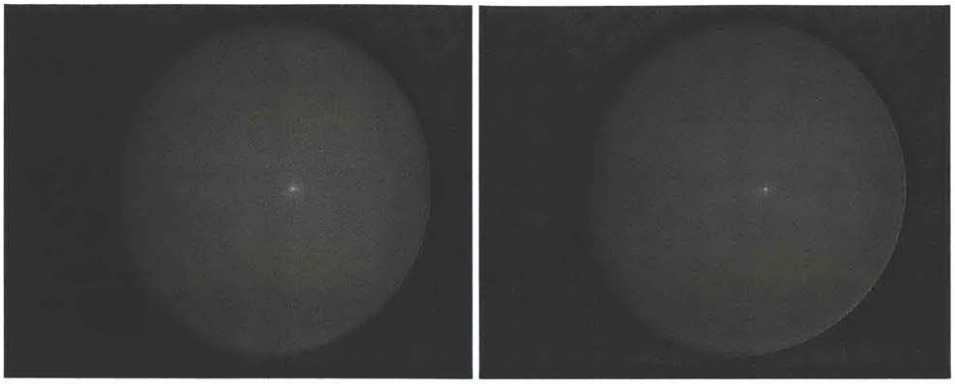

Figure 3-6: *Two log images of a sphere (alumina oxide). A real image is shown on the left. A synthesized image using tabulated BRDF data is shown on the right.*

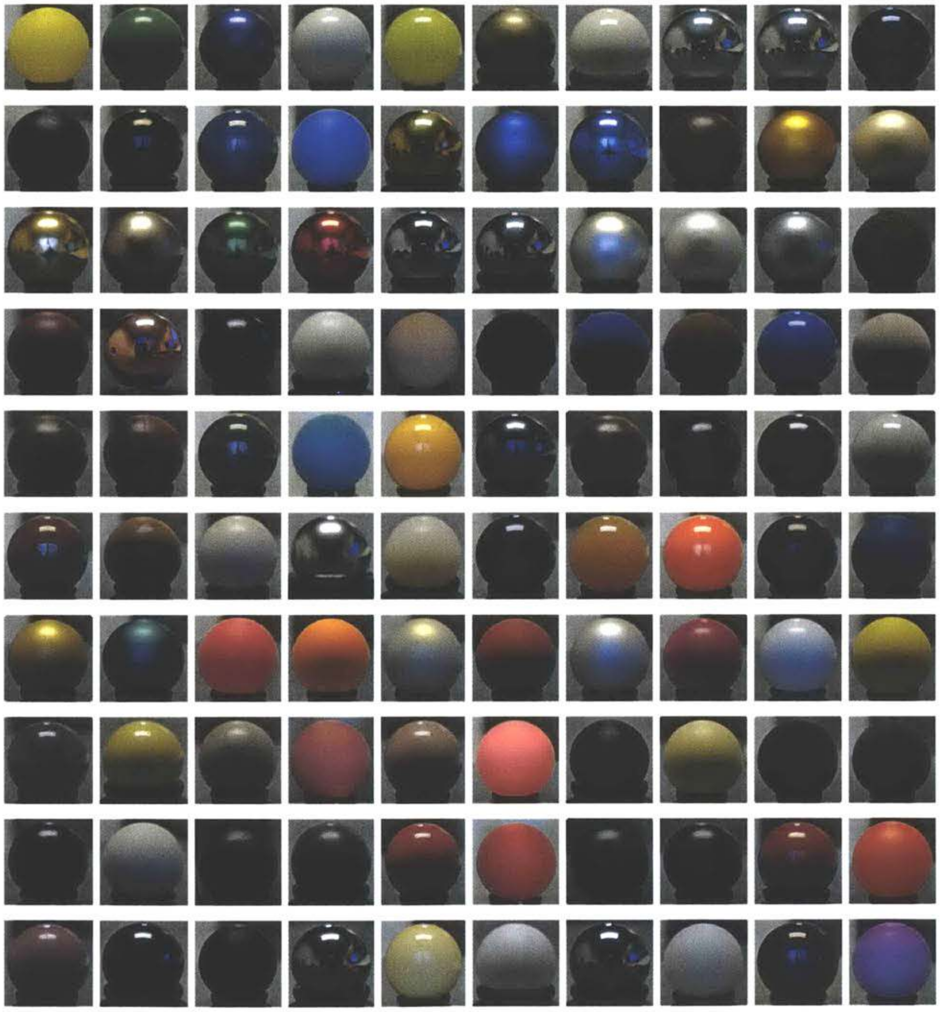

Figure 3-7: *Pictures of 100 of the acquired materials.*

3.7 Comparison with Analytic Reflectance Models

Each of the acquired BRDFs can be used to estimate the parameters of the analytic reflectance models presented in chapter 2. In this section, I describe how to fit parametric models to my acquired data and the difficulties associated with it. I also present a comparison of the measured reflectance with the results of the fitting to the analytic models.

I have fit BRDF measurements of four different materials from my dataset to two different analytic reflectance models – the Ward model and the single-lobe Lafortune model. I did not fit my data to the popular Phong model since it is not physically plausible (e.g., it violates energy conservation), but the Lafortune model can be viewed as a generalized version of the Phong model. Both the Ward and Lafortune models have been used previously [55, 25, 29] to represent measured BRDFs. Since these models are non-linear expressions the fitting process requires performing non-linear optimization. I use the Matlab implementation of the Levenberg-Marquardt algorithm [31].

There are three major shortcomings when fitting the measurements to the analytic reflectance model. First, I note that it is not computationally possible to perform the non-linear optimization based on all entries of a tabulated BRDF (each tabulated BRDF has over 4 million values). Thus, the optimization is performed based on only 6000 samples (2000 samples for each color channel). The second shortcoming of the fitting procedure is the choice of the error function used in the non-linear optimization. The error function is a very important factor since BRDFs have a wide range of values – the peaks have typically a few orders of magnitude higher values than the diffuse component. The error expressed in the original space tends to overemphasize the importance of the BRDF peaks and deemphasize the off-peak values. Instead, I use the error calculated in log space. While this error metric is not ideal, it produces perceptually better results. Deriving a perceptually optimal error metric remains an open research problem. The third problem is the sensitivity of the non-linear optimization to the quality of the initial guess. I found that the results are usually not the same if the initial guess is different. When the initial parameter values are too far from the optimal values, the optimization might not converge to the global minimum.

I found that the Ward model performed consistently worse than the Lafortune model.

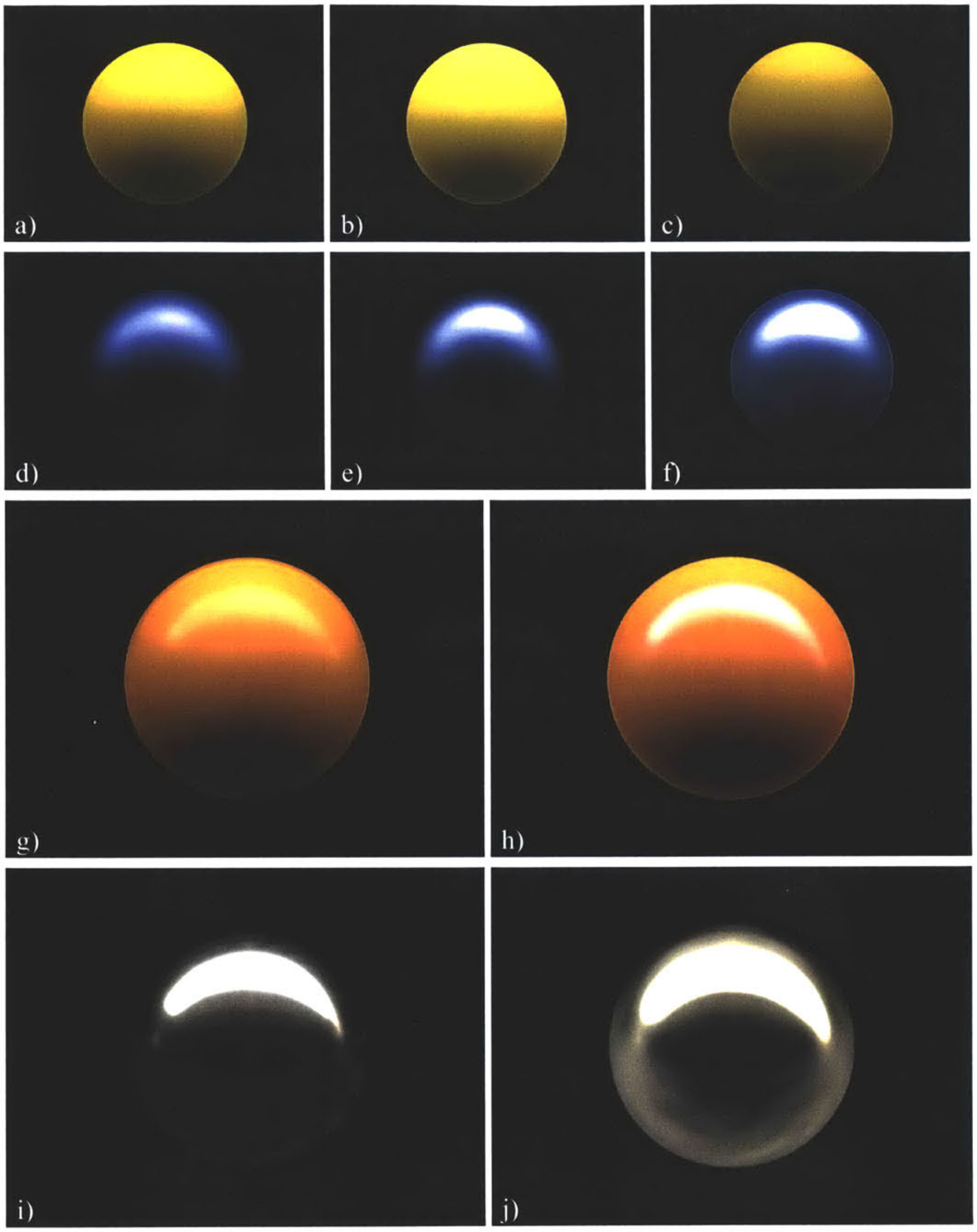

Figure 3-8: *Comparison between analytic models and measured reflectance. The first row shows a yellow paint sphere under complex illumination rendered from raw tabulated data (a), the Lafortune model (b), and the Ward model (c). The second row shows a blue metallic sphere rendered from raw tabulated data (d), the Lafortune model (e), and the Ward model (f). The third and fourth rows show orange plastic and nickel spheres rendered from raw tabulated data (g, i), and Lafortune model only (h, j).*

This is illustrated in Figure 3-8. This result might be due to the fact that the Ward model is not capable of representing Fresnel effect (increasing specularity at grazing angles). I also observe that the quality of the fit to the Lafortune model varies among BRDFs. The yellow paint BRDF fits the model the best, while the nickel BRDF has the worst fit. I also observed that the fitted lobes are consistently larger and brighter than the measured lobes (the rendered images appear consistently brighter in the specular regions). The errors of the fits calculated in log space for the four BRDFs are as follows: yellow-paint: 5%, blue metallic paint: 10%, orange plastic: 12%, and nickel: 21%. There is a clear perceptual difference between the pictures of a sphere rendered using analytic reflectance models and measured reflectance data (Figure 3-8). This implies that using densely tabulated reflectance data directly for rendering offers significant advantages.

3.8 Summary

In this chapter, I have described the BRDF measurement device that I have developed for the rapid acquisition of dense BRDF measurements. Even though each material sample requires over six hours to capture, analyze, and rebin, this process is still orders of magnitude faster than previous documented systems. As a result, I was able to acquire more than 100 dense BRDFs from a wide range of different surface materials.

The measured BRDFs can be used directly in computer graphics rendering applications. I used Dali, a global illumination renderer, developed by Henrik Wann Jensen. Several examples are shown in Figure 3-9, where a teapot is rendered under natural illumination[2] using the raw acquired data. (The details of of the BRDF shader used in these renderings are described in the appendix A.) Suppose, however, that one wants to render a material that is not one of the original materials. One option is to find a spherical sample of the desired material and measure its reflectance, using the methods described in previous sections. A second, more attractive option is to synthesize the reflectance of this material by combining the reflectances of the original measured materials from my database. This leads to the construction of the data-driven reflectance model that is described in the next chapter.

---

[2] The light probe is obtained from www.debevec.org.

Figure 3-9: *Rendered teapots using BRDFs from my database: nickel, hematite, gold paint, pink fabric, dark red paint, fruitwood-241, oxidized steel, polyurethane, violet rubber, pink jasper, brown wax, blue metallic paint, yellow paint, aluminum bronze, and green acrylic.*

Chapter 4

Model Construction

This chapter describes the construction of a data-driven reflectance model. The construction process uses machine learning techniques and, in particular, unsupervised learning methods. The classic problem in unsupervised learning is to infer a function from a set of example outputs. Each acquired isotropic BRDF can be treated as a vector in a high dimensional space ($\mathbf{y} \in \mathbb{R}^{\mathrm{D}}$) – each tabulated 3D isotropic BRDF is unrolled to form a D-dimensional vector. Only a subset of all possible vectors $\mathbf{y} \in \mathbb{R}^{\mathrm{D}}$ corresponds to valid isotropic BRDFs. In order to construct a data-driven reflectance model I attempt to find a description of this subset based only on a relatively small number of training vectors. Dimensionality reduction methods are used to solve this problem. This analysis assumes that all BRDFs corresponding to real-world materials lie on a relatively low dimensional manifold embedded within the high-dimensional space $\mathbb{R}^{\mathrm{D}}$. This is a common assumption used by others [9] and it is consistent with the relatively small number of parameters seen in analytic BRDF models.

First, I use Principal Component Analysis (PCA) to discover a linear subspace (a low dimensional hyperplane embedded in the high dimensional space) that best describes my dataset (in a least-squares sense). I discuss the validity of the linear subspace model and the problems associated with it. Next, I analyze the dataset using non-linear dimensionality reduction techniques that are more general than the linear analysis methods since they are capable of discovering curved low-dimensional manifolds embedded in high dimensional spaces. Once I obtain the description of the manifold (or the linear subspace) for isotropic

BRDFs I can impose a parameterization of the discovered manifold and synthesize new plausible isotropic BRDFs by interpolating between points on this manifold. Thus, I obtain a generative data-driven BRDF model.

4.1 Analysis Using PCA

The first isotropic BRDF modelling approach that I will explore is a linear model. Finding a linear model involves computing an orthogonal basis vector set of a specified dimension that best represents the BRDFs in my dataset. In this case, the analysis tools for both manifold discovery and interpolation are well known. Principal Component Analysis, described in detail in chapter 2, effectively determines a set of basis vectors that span the desired subspace. Linear combinations of these basis vectors, sometimes called principal components or directions, can be used for approximation and interpolation within the space of given samples. Principal Component Analysis has proven extremely effective in several problem domains, such as face synthesis [3] and radiance interpolation [6]. Potential linear manifolds are generally suggested when there is a noticeable plateau in the magnitudes of the ordered eigenvalues associated with each principal direction. When this plateau occurs on the k^{th} eigenvalue, we can model the data as a k-dimensional linear subspace with a residual error bounded by the square root of the sum of the squares of the remaining eigenvalues. Independent of any plateau, however, the best k-dimensional approximation (under a least-squares metric) is given by a linear combination of the fist k principle directions.

I began the analysis of the BRDF samples by searching for a linear embedding manifold (a hyperplane). In this approach each of 104 BRDF samples is treated as a column vector to form a 4,374,000 by 104 measurement matrix $\mathbf{Y}$. Initially I performed the analysis in the linear domain, but the reconstructed BRDFs exhibited large errors in the non-specular areas. Upon inspecting the data I found that the specular reflections were several orders of magnitude larger than values in areas of non-specular reflection, and errors in these regions dominate under a least-squares metric (e.g., measurement errors in the specular regions have larger values than the values in the off-specular areas). Therefore, I have decided to perform the analysis in the log space. (I apply the natural logarithm to each element of vector $\mathbf{Y}$.) This operation flattens the range of the BRDF values. It increases the importance of the reflectance in the off-specular regions – these areas are also perceptually very important. The operation can be also justified by the fact that the human visual system is more sensitive to ratios of radiance values rather than absolute radiance values.

In order to compute the principal components $\mathbf{U}$ and the coordinates $\mathbf{X}$ of the original BRDFs in the new basis, the following steps are performed. First, the data points $\mathbf{Y}$ have to be mean centered (the mean of all log-BRDFs is computed and subtracted from $\mathbf{Y}$). Next, the matrix $\mathbf{B} = \mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ is computed by taking dot products between all pairs of the BRDF vectors. Note that the matrix $\mathbf{B}$ is symmetric; thus, the number of computations is halved. Moreover, $\mathbf{B}$ is only a 104 by 104 matrix. Then, the eigenvalue decomposition of the matrix $\mathbf{B}$ is performed (e.g., using `eig` function in Matlab):

$$\mathbf{B} = \mathbf{Y}^{\mathrm{T}}\mathbf{Y} = \mathbf{V}\Lambda_{\mathbf{B}}\mathbf{V}^{\mathrm{T}}, \tag{4.1}$$

where $\mathbf{V}$ is the orthonormal matrix of eigenvectors and $\Lambda_{\mathbf{B}}$ is the diagonal matrix of decreasing eigenvalues. The principal components $\mathbf{U}$ of the matrix $\mathbf{Y}$ can then be computed using the following formula:

$$\mathbf{U} = \mathbf{Y}\mathbf{V}\Lambda_{\mathbf{B}}^{-\frac{1}{2}}. \tag{4.2}$$

The coordinates $\mathbf{X}$ of original BRDFs in the new basis are computed efficiently from:

$$\mathbf{X} = \Lambda_{\mathbf{B}}^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}. \tag{4.3}$$

In this way I obtain $\Lambda_{\mathbf{B}}$, $\mathbf{U}$, and $\mathbf{X}$. A plot of the eigenvalues $\Lambda_{\mathbf{B}}$ sorted in order of decreasing magnitude the corresponding reconstruction error for the BRDF database is shown in Figure 4-1. I provide a visualization of the first principal components in Figure 4-2. Furthermore, in Figure 4-3 I show the reconstruction of a few different BRDFs using first 5, 10, 20, 30, 45, 60, and all principal components. Good reconstruction is usually obtained using the first 30-40 components. Note that there is a considerable fall off in the sequential values seen in this plot but there is no noticeable plateau. However, the residual noise reaches a level that is comparable to the estimated measurement noise around the 45^{th} eigenvalue. This dimension of the embedding subspace is considerably higher one would assume based on the typical number of parameters used in analytic BRDF models.

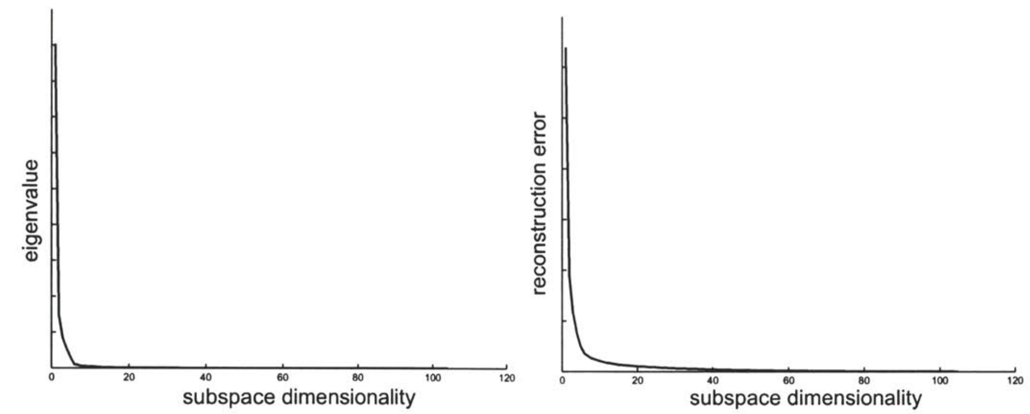

Figure 4-1: *Plot of the eigenvalues (left) and the corresponding reconstruction error as a function of dimensionality (right).*

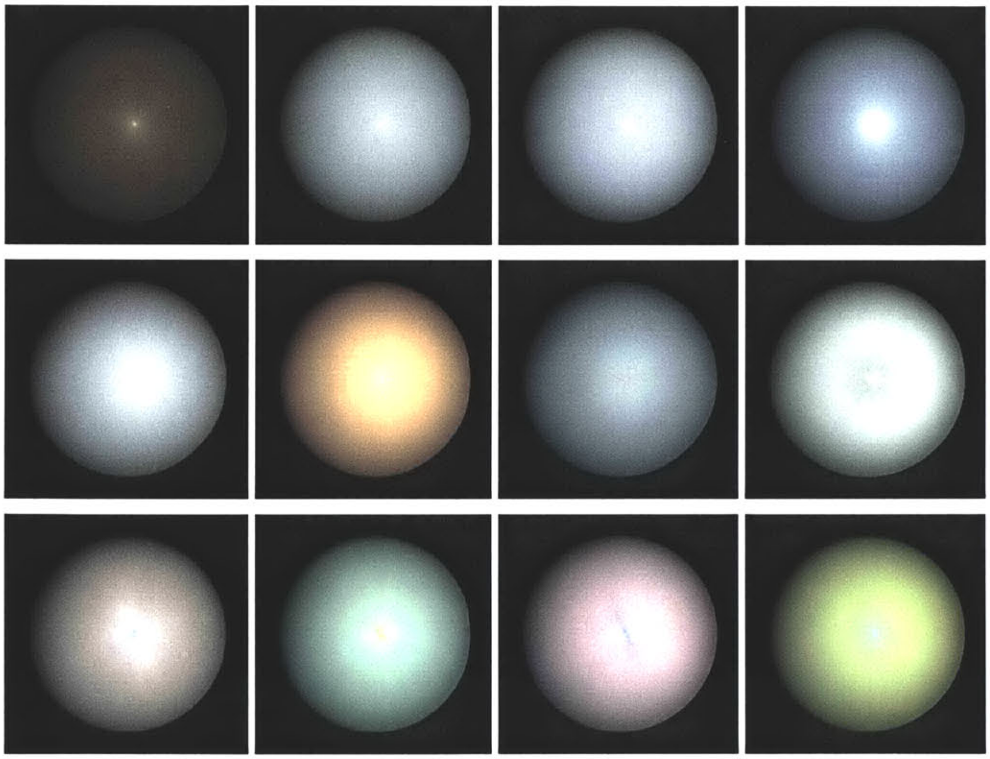

Figure 4-2: *The mean and the first 11 principal components. In order to visualize each principal component, its minimum value is subtracted first. Next, a sphere is rendered under a point light source using the principal component as a BRDF.*

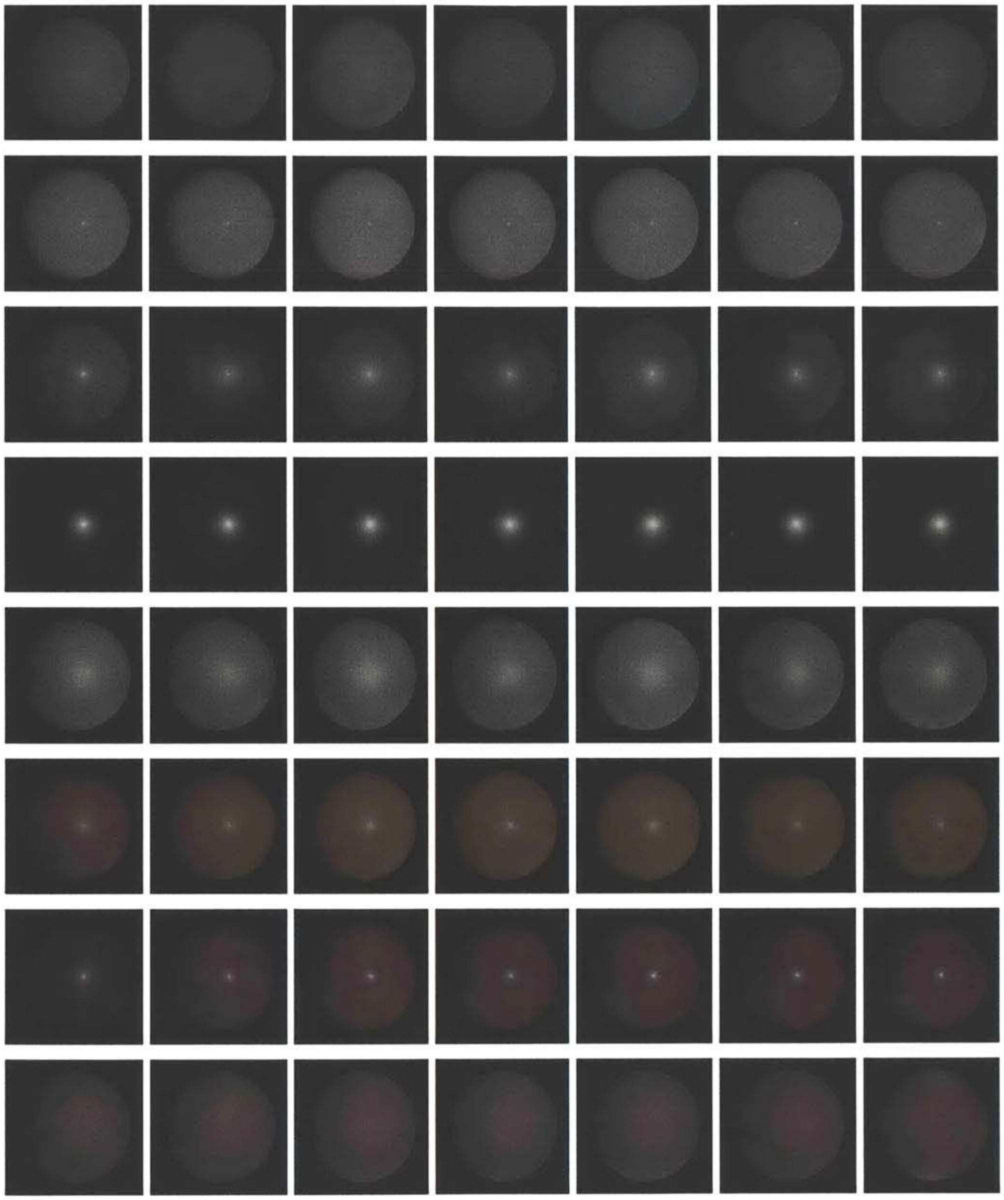

Figure 4-3: *Reconstruction of different BRDFs from principal components in order (left to right) of increasing number of components: 5, 10, 20, 30, 45, 60, and all. Each row represents one BRDF: blue rubber, alumina oxide, violet acrylic, nickel, pearl paint, yellow matte plastic, red phenolic, and pink fabric.*

Nevertheless, it is easy to verify that the 45-dimensional space defined by the first principal components reconstructs all our measured BRDFs well, as shown in Figure 4-3.

4.2 Analysis Using Other Linear Methods

Principal Component Analysis computes the optimal orthogonal linear basis such that the sum of the squared errors is minimized. Otherwise, it places no constraints on the basis vectors. I will propose two other potential ways of deriving the reflectance model using linear analysis. However, it is impossible that any of these methods could achieve a more compact (low-dimensional) representation than PCA under the least-squares error metric. Therefore, while these methods might be useful in specific applications, they will still exhibit higher dimensionality than one would assume.

4.2.1 Analysis Using Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) also attempts to compute the optimal basis $\mathbf{U}$ and the coordinates $\mathbf{X}$ for the matrix $\mathbf{Y}$. The analysis constrains $\mathbf{U}$, $\mathbf{X}$, and $\mathbf{Y}$ to be non-negative. All BRDFs that comprise $\mathbf{Y}$ are by default non-negative. This form of factorization has an appeal since each BRDF must be non-negative to be physically valid. A BRDFs is a description of a strictly non-negative process since it assumes that all incident radiance is reflected in some direction and that reflected radiance is always positive. Therefore, the reflectance model obtained using the basis BRDFs derived from NMF would preserve the non-negativity property of the synthesized BRDFs. This is not the case for the linear model derived using PCA (a given linear combination of principal vectors must be checked explicitly to verify whether the synthesized BRDF is non-negative). The only remaining property that needs to be checked for the NMF-derived model is energy conservation. The use of NMF is suitable in the case where one might expect that the space of BRDFs is described by a linear combination of some "inherent" basis set of primal BRDFs (e.g., diffuse, specular BRDF).

Unfortunately, NMF does not have a closed-form solution. It requires an iterative procedure that usually takes a long time to converge. Furthermore, in order to be implemented efficiently the whole matrix **Y** needs to be stored in the main memory. However, the size of the matrix **Y** is at least 2GB – slightly too big for current desktop systems. This makes the implementation too slow on current systems. Analysis using NMF is therefore left as future work.

4.2.2 Analysis Using Multidimensional Scaling

The second alternative to PCA is to use the multidimensional scaling (MDS) with a user-defined perceptual distance matrix to analyze the space of BRDFs. If a Euclidean distance metric is used then MDS is equivalent to PCA. However, one can argue that the Euclidean distance metric[1] for BRDFs is not perceptually meaningful. An alternative approach is to use human experts to define a meaningful perceptual metric. This can be done by estimating pairwise distances between all BRDFs in the dataset. However, even for a relatively small dataset this would be a difficult task – one expert would have to estimate $(104 \times 103)/2 = 5356$ pairwise distances between BRDFs. I note that the task could be greatly simplified since experts could estimate the similarity of the physical material samples that are used in the measurement process rather than the similarity of the rendered pictures. In order to obtain a robust perceptual metric, the estimation process should be performed by many experts and the final distance matrix should be averaged or otherwise combined.

Once the perceptual distance matrix is computed, MDS could be used to compute the dimensionality of the embedding and the corresponding embedding coordinates of each BRDF as described in chapter 2. While MDS is very useful in analyzing the dimensionality of the data, the algorithm does not provide the backprojection function that would allow synthesizing new BRDFs. Therefore, this procedure does not provide a useful reflectance model. Furthermore, this approach has been used by Pellacini et al. [40] to derive a psychophysical light reflectance model. (Their analysis is based on rendered pictures using an analytic reflectance model and not measured data.) Thus, I decide not to pursue this approach.

---

[1] As previously noted, I perform PCA on the logarithm of BRDFs; therefore, even for my PCA the distances are not Euclidean.

4.3 Limitations of the Linear Model

The model obtained using Principal Component Analysis has some drawbacks for practical use. The most serious is the large number of principal vectors – users need to specify at least 45 different parameters in order to span the whole BRDF space. Furthermore, the space spanned by the 45 principal vectors includes vectors that correspond to unlikely or physically implausible BRDFs, as I will demonstrate. I start by considering the combinations of BRDFs outside the convex hull of the original BRDFs. Since the subspace extends infinitely, some of the vectors outside the convex hull will produce physically implausible BRDFs (e.g., the non-negativity or energy conservation property is not met).

Determining the validity of BRDFs within the convex hull of measured data samples is a much more difficult undertaking. First, I will show that conceptually all convex combinations of the source BRDFs would produce physically valid BRDFs (non-negativity, energy conservation, and reciprocity properties are preserved). To demonstrate this point I describe a thought experiment that explains how to "manufacture" a material with a BRDF equivalent to any convex combination of the source BRDFs. First, consider that a BRDF is defined with respect to an infinitesimally small surface element, but in practice sensors integrate over finite surface patches when measuring reflectance. Let's imagine a specific surface patch that is viewed at exactly one pixel of the sensor. If we assume that this surface patch is composed of tiles, and each of these tiles can be made of a different base material, then one could design material with a reflectance equivalent to any convex combination of the source BRDFs by assigning appropriate (possibly dithered) material to each tile. This construction process is illustrated in Figure 4-4. While it would be possible to "manufacture" these materials, not all of them appear "natural." This is illustrated in Figure 4-5 in which I show BRDFs that are convex combinations of the source BRDFs – these BRDFs correspond to unlikely materials. In this thesis I focus on modelling reflectance of real-world materials. Therefore, these limitations lead me to analyze the dataset using non-linear dimensionality reduction techniques.

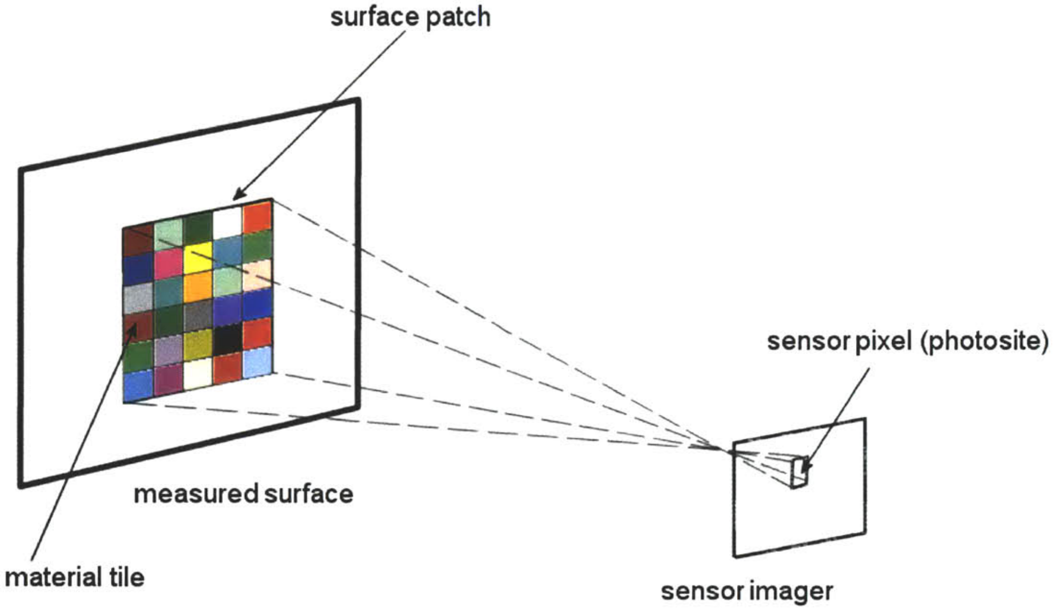

Figure 4-4: *"Manufacturing" a material with a BRDF equivalent to any convex combination of the source BRDFs.*

4.4 Non-linear Dimensionality Reduction

The discussion in the previous section, combined with the relative effectiveness of many low-parameter count analytic BRDF models, suggests that the space of BRDFs corresponding to materials found in nature lies on a lower-dimensional non-linear manifold that is embedded in the 45D linear subspace. I have applied recently developed non-linear dimensionality reduction techniques to discover this lower dimensional manifold. I presented the overview of the non-linear dimensionality reduction techniques in chapter 2. In order to construct an effective data-driven model one has to compute a continuous mapping between the embedding and the original (ambient) space. Thus far only two methods are capable of providing this functionality: Automatic Alignment [48] and charting [5].

I chose to use charting because it is explicitly designed to work well with a small numbers of samples and to suppress measurement noise, two conditions that tend to break methods for dimensionality reduction from local relationships. For charting, one must specify a set of chart centers, a width parameter σ for the Gaussians, and a target dimensionality d.

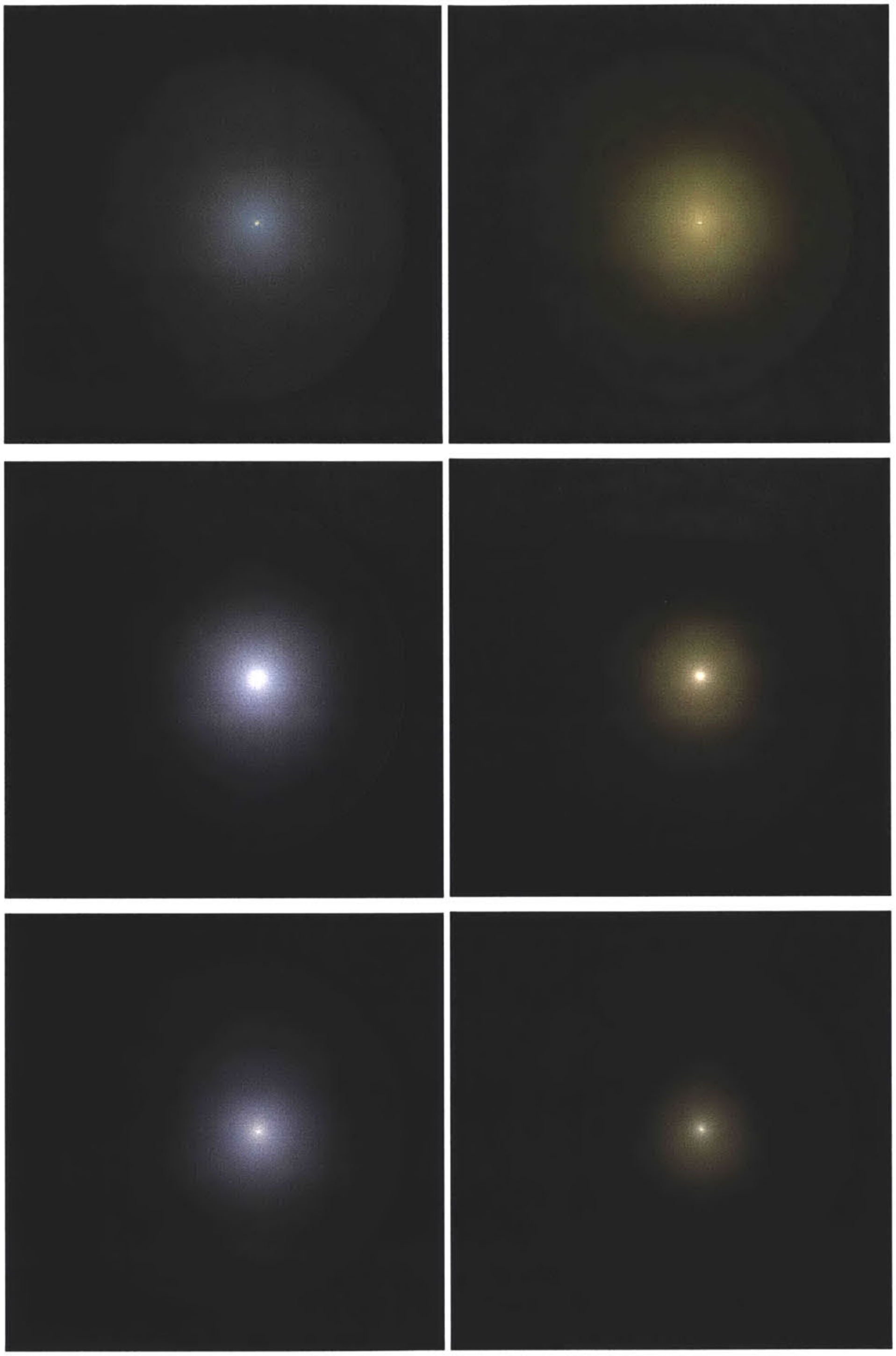

Figure 4-5: *Convex combinations that correspond to unlikely BRDFs: combinations of narrow and wide lobes (first row), combinations of wide and medium size lobes (second row), combinations of narrow, wide, and medium size lobes (third row).*

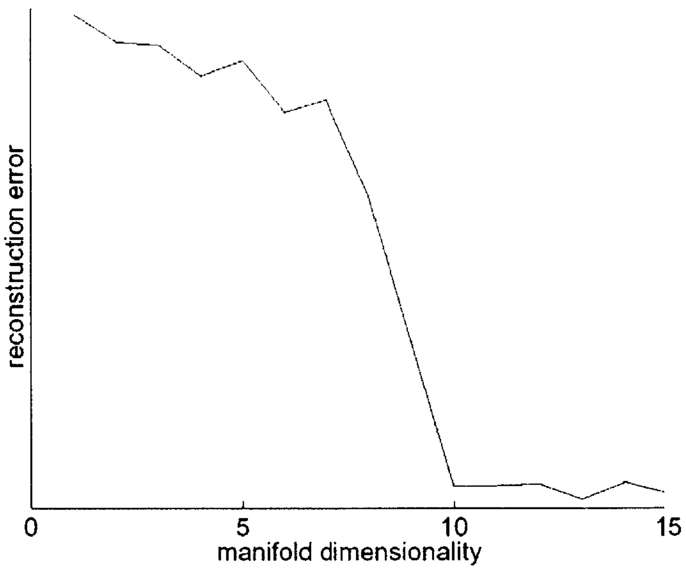

Figure 4-6: *Data reconstruction error as a function of the dimensionality of the global chart. There is a sharp drop in the residual error for 10D embedding space. However, at 10D some visual artifacts are still observed in the reconstructed BRDFs. These artifacts are not present at 15D when the residual error drops below 1%.*

I used the default settings: one chart centered on each data point and σ = half the average distance between each point and its closest neighbor. Note that locating a chart on a point does not cause the manifold to pass through that point – only near it.

As with PCA, the data-reconstruction error of a charted dataset gives an indication of the true dimensionality of the manifold. Figure 4-6 shows a sharp drop in the residual error at 10D embedding space. However, when using 10D embedding space some of the reconstructed BRDFs have visual artifacts. These artifacts are not present at 15D when the residual error drops below 1% (comparable to reconstructions with 45 principal components). Observe that the reconstruction error does not decline monotonically because each dimensionality may merit a different flattening. For example, if the data were sampled from a truncated cone, the best 1D chart would simply be height along the cone, while the best

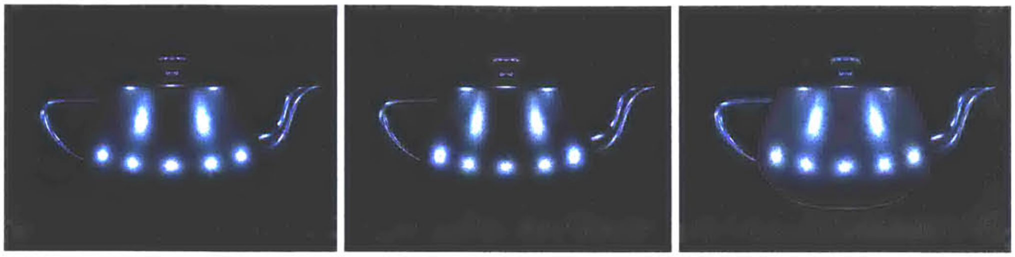

Figure 4-7: *Non-linear spaces generate valid BRDFs where linear spaces fail. Original BRDF corresponding to a point A on a* 45^{th} *dimensional hyperplane (left). Physically not plausible reflectance (hole in the middle of the specular highlight) corresponding to moving away from a point A on the* 45^{th} *dimensional linear subspace (center). Physically plausible reflectance corresponding to moving equally far away from point A on the* 15^{th} *dimensional non-linear manifold (right).*

2D chart would flatten the cone into an annulus. Each flattening would suppress the noise in different directions, some more fortuitous than others.

A charted manifold of BRDF data makes it possible to treat the space of BRDFs as if it were linear, and to identify meaningful axes of variation in this embedding space. An interpolating or extrapolating line in this space is a non-linear curve in the original BRDF space that passes closer to the data density that the equivalent straight line would (on average), simply because it stays on the manifold where a straight line does not. This translates directly to superior BRDF synthesis (Figure 4-7).

4.5 Obtaining More Data Points

The current number of data points (BRDFs) is relatively small for the non-linear dimensionality reduction methods. Ideally, the manifold should be sampled densely. Obtaining a large number of BRDFs even using my fast device poses some difficulties: first, homogeneous spherical samples of these materials must be available; second, it is surprisingly difficult to find new materials that provide additional information (i.e., materials that either the linear or non-linear model cannot accurately estimate). As a result, adding new materials ad-hoc does not necessarily increase the useful range of the data-driven model. I propose two alternative approaches to obtain more valid BRDFs.

4.5.1 Efficient Measurement

One obvious way to obtain more valid BRDFs is to measure the reflectance of other materials. Fortunately, when using the assumption that all BRDFs lie on the 45D linear subspace, the sampling process can be greatly simplified. Since the basis vectors for this subspace are already computed, we need only to estimate the 45D coordinates of a BRDF in the subspace – each new BRDF can be represented using just 45 floating point numbers. In chapter 6 I will describe a method for estimating these coefficients using a few hundred BRDF samples. This implies the measurement process can carried out efficiently using gonio-refelectometers that do not require spherical material samples.

4.5.2 BRDF Hallucination

The second method to generate more data points that correspond to plausible BRDFs involves a user in the loop. It is possible to use linear combinations of the source BRDFs (or linear combinations of 45 principal components) to generate more data points, and better define the non-linear manifold estimate. As noted previously, some of these data points correspond to likely BRDFs but many of them do not. A material expert could be used to determine whether a given linear combination is a valid BRDF. Humans can easily judge the validity of a BRDF using three-dimensional renderings. To facilitate this, an object with a given linear combination of BRDFs could be rendered under both complex natural illumination and single-point source illumination. With this approach thousands of additional data points could be easily synthesized, and these additional data points could be used to help to the manifold specification.

4.6 Discussion

Since the number of BRDFs in my dataset is relatively small, it is not possible to draw strong conclusions about the dimensionality of the space of BRDFs for all real-world materials. However, my data indicates that the linear subspace describing all isotropic BRDFs has **at least** 45 dimensions. Similarly, my data indicates that the non-linear manifold describing all BRDFs found in nature has **at least** 15 dimensions.

However, the dimensionality of the linear subspace for all isotropic BRDFs is probably not much larger than 45. In order to strengthen this claim I perform cross-validation experiments described in more detail in chapter 6. I remove four BRDFs that look different from the others in the dataset. Next, I approximate these four BRDFs as linear combinations of the remaining 100 BRDFs. (Alternatively I perform PCA on just 100 BRDFs to derive 45 new principal components; then I approximate the four BRDFs as linear combinations of the new principal components.) I found that these linear combinations approximate these four removed BRDFs to within measurement error.

It is worthwhile to mention that the discrete representation used for each BRDF imposes some limitations on the BRDFs that can be generated using my data-driven model. For example, the spacing between the sampling bins determines the maximum frequency contents of the generated BRDFs. Therefore, a perfectly specular (mirror) BRDF would be impossible to synthesize using the data-driven model because it is equivalent to the Dirac delta function that would require an infinite number of discrete bins.

Chapter 5

User-Defined Parameterization

The major limitation of both the linear and the non-linear data-driven model described in the previous chapter is that the resulting basis vectors often have no intuitive meaning. In order to address this problem I have developed a modelling approach that simplifies the design and exploration of new BRDFs. I investigated methods for characterizing material traits by analogies derived from the existing samples. I believe that such methods provide the best and most intuitive user interface [40].

My model is built from actual physical measurements and it reproduces these measurements. Moreover, modelling parameters have been determined for a large collection of real-world materials – materials I have measured. I believe that the most useful BRDF design paradigm would be one where users can choose as a starting point some type of the material, similar to the one they desire, and then specify changes to this material. In the case of a data-driven model they can pick any of the measured materials as a starting point. Then, they would change the reflectance properties of this material according to one of the two schemes. (These navigation modes are applicable for both linear and non-linear manifold models.) The simplest method is to choose another BRDF and move in this direction. Although of limited use, this method works well for perceptually similar materials. A more useful approach is to define directions corresponding to a desired trait (the parameterization direction is a 45D vector for linear space and a 15D vector for non-linear space). Then users can modify a BRDF by adding trait vectors to the current position to increase the trait, or subtracting it to decrease the trait. It is important to note that they can always backproject the current point onto the original BRDF space to check the corresponding BRDF. In the next section I describe various procedures for identifying trait vectors.

5.1 Trait Vectors Specification

My modelling approach requires users to specify a sufficient set of trait vectors to span the space of the particular model (linear subspace or non-linear embedding space). These vectors will be associated with corresponding parameters of the data-driven model. In order to specify a trait vector, users classify some number of BRDFs according to the presence or absence of the trait. For example, each acquired BRDFs is assigned a class label: if the trait is present then the class label value is 1 ($y = 1$); if the trait is not present then the label value is -1 ($y = -1$). I also allow the user to leave a BRDF unspecified in cases where the trait is hard to determine or simply does not apply. Usually the more samples the user specifies for each class, the more precise the direction is. There are many different ways to define the parameterization directions based on the classification. I have examined three different methods: mean difference, Fisher's linear discriminant, and support vector machines.

5.1.1 Mean Difference

The mean difference is the simplest of the three approaches. It has been used before to specify user-defined parameters in the space of 3D face models [3]. Let x_i^1 denote the coordinates of the i^{th} BRDF in the embedding space for which the trait is present (in case of the linear analysis these are the coordinates in the linear subspace). Similarly, x_i^2 denotes the coordinates of the i^{th} BRDF for which the trait is not present. First I compute the means of the coordinates in each class. Then to form the trait vector $\mathbf{w}$ I subtract the means from each other:

$$\mathbf{w} = \frac{1}{N}\sum_{i=1}^{N} x_i^1 - \frac{1}{M}\sum_{i=1}^{M} x_i^2. \tag{5.1}$$

The vector $\mathbf{w}$ is the parameterization direction. This direction vector is then scaled and applied (added or subtracted) to the current point in the embedding space.

5.1.2 Fisher's Linear Discriminant

The second method for specifying trait vectors uses Fisher's linear discriminant [14]. Assume that the data points $x_i \in \mathbb{R}^d$ are divided into two sets $X_1 = \{x_1^1, ..., x_n^1\}$ and $X_2 = \{x_1^2, ..., x_m^2\}$ and let $X = X_1 \cup X2 = \{x_1, ..., x_{n+m}\}$. Let μ_1 and μ_2 define the means of the data points in each set:

$$\mu_1 = \frac{1}{n}\sum_{i=1}^{n} x_i^1 \tag{5.2}$$

$$\mu_2 = \frac{1}{m}\sum_{i=1}^{m} x_i^2. \tag{5.3}$$

Fisher's linear discriminant attempts to find a direction vector **w** such that two objectives are optimized: (1) The projections of the means μ_1 and μ_2 onto the vector **w** are the most separated – the following expression needs to be maximized:

$$\frac{1}{|\mathbf{w}|}\mathbf{w}^T S_B \mathbf{w}, \tag{5.4}$$

where S_B is defined as:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T. \tag{5.5}$$

(2) The data points projected onto **w** are clustered near their corresponding means – the sum of variances of each class around its mean projected onto **w** is minimized:

$$\frac{1}{|\mathbf{w}|}\mathbf{w}^T S_W \mathbf{w}, \tag{5.6}$$

where S_W is defined as:

$$S_W = \sum_{i=1}^{2} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T. \tag{5.7}$$

Optimizing these two constraints is equivalent to maximizing the value of $J(\mathbf{w})$:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}. \tag{5.8}$$

The optimal solution for w is obtained using the formula:

$$\mathbf{w} \cong S_W^{-1}(\mu_1 - \mu_2). \tag{5.9}$$

The vector **w** computed in this way becomes the trait vector.

5.1.3 Support Vector Machines

In this section I give an overview of the third method of computing trait vectors – the support vector method [53]. This method finds the optimal hyperplane that separates two sets of vectors. Assume a set of vectors x_i, where each vector has an associated label $y \in \{-1, 1\}$ (the label separates the vectors into two subsets):

$$(x_1, y_1), ..., (x_n, y_n), x_i \in \mathbb{R}^d, y \in \{-1, 1\}.$$

Assume that the vectors in the two classes are linearly separable – it is possible to find a hyperplane specified by a vector $\mathbf{w}$ and a constant b such that:

$$y_i[\mathbf{w}^T x_i + b] \geq 1, i = 1, ..., n. \tag{5.10}$$

In the support vector method optimality of the hyperplane is specified in terms of a margin – a distance from the hyperplane to the closest example in either set. The optimal hyperplane is the one with the maximal margin and is unique. The scale of the vector $\mathbf{w}$ and the constant b is arbitrary, but assume the scale is fixed such that for the points closest to the hyperplane:

$$y_i[\mathbf{w}^T x_i + b] = 1. \tag{5.11}$$

It can be shown that to compute the optimal value of $\mathbf{w}$ and b one needs to solve the following optimization problem:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\mathbf{w}^T\mathbf{w} \\ &\text{subject to} && y_i(\mathbf{w}^T x_i + b) \geq 1, i = 1, ..., n. \end{aligned} \tag{5.12}$$

The optimal values of $\mathbf{w}$ and b can be also computed by introducing Lagrange multipliers α_i and solving the following problem:

$$\begin{aligned} &\text{minimize} && J(\mathbf{w}, b, \alpha) = \|\mathbf{w}\|^2/2 - \textstyle\sum_{i=1}^{n} \alpha_i(y_i[\mathbf{w}^T x_i + b] - 1) \\ &\text{subject to} && \alpha_i \geq 0, i = 1, ..., n, \end{aligned} \tag{5.13}$$

with respect to **w**, b. The minimum value of the objective function has the partial derivatives equal to zero:

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i x_i = 0, \tag{5.14}$$

$$\frac{\partial}{\partial b} J(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{5.15}$$

Then using equation 5.14 to solve the optimization problem one obtains the following maximization problem:

$$\begin{aligned} \text{minimize} \quad & J(\alpha) = \textstyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\ \text{subject to} \quad & \alpha_i \geq 0, i = 1, \ldots, n, \\ & \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{5.16}$$

This optimization problem can be solved efficiently using quadratic programming. The non-zero values of Lagrange multipliers α_i correspond to support vectors – the vectors in both datasets that define the hyperplane. The vector **w** can be computed using equation 5.14, or even simpler using:

$$\mathbf{w} = \sum_{i \in SV} \alpha_i y_i x_i. \tag{5.17}$$

The vector **w** defines the normal to the hyperplane separating the two subsets and it is used as the trait vector. In practice the vectors computed using the support vector method gave the most reliable trait vectors.

5.2 Enforcing Physical Validity of the Data-Driven Model

I would like to ensure that outputs from the data-driven model are physically plausible. This means that the model has to, at least, preserve three basic principles of physics: non-negativity, reciprocity, and energy conservation. The details of each of the principles are described in subsection 2.1.3. In general I have to disallow movements on the manifold that do not adhere to these principles. I consider each of the principles in order:

- **Non-negativity:** I allow the user to move only in the space so that all the values in the backprojected vector are positive.

- **Reciprocity:** As mentioned before, reciprocity in the data-driven model is assured by construction since I store only half of the BRDF vector, and the other half is constructed by symmetry.
- **Energy conservation:** I enforce this property by not allowing the users to produce BRDFs for which the sum of energy reflected from the surface is larger than the incident energy. (This has to be true for all incident light directions.)

In chapter 4 I have shown that it is possible to physically construct a material with the BRDF equivalent to any convex combination of source BRDFs. It follows that any convex combination of the source BRDFs is physically plausible – it satisfies the three principles. Therefore, if a synthesized BRDF is inside the convex hull of the source BRDFs then it meets these principles and it is not necessary to explicitly check them.

5.3 Modelling Results

I have shown how to construct the perceptually meaningful parameters and how to ensure that the synthesized BRDFs are physically plausible. This section presents the results from a typical model construction session.

A test subject was asked to characterize each of the BRDFs from my database using 16 different traits. These included *redness*, *greenness*, *blueness*, *specularness*, *diffuseness*, *glossiness*, *metallic-like*, *plastic-like*, *roughness*, *silverness*, *gold-like*, *fabric-like*, *acrylic-like*, *greasiness*, *dustiness*, *rubber-like*. In a sense, these parameters are arbitrary since the classification is completely based on the subject's interpretation. I could have chosen traits without physical connotations, such as *ugly* or *pleasing*. Alternatively, the traits could have been based on actual measurable quantities, such as conductivity and mean surface variation. My test subject characterized each BRDF as one of three choices: 1) possessing the particular trait, 2) not possessing the trait, or 3) unclear.

Next the subject's characterizations were used to build a model in both the linear and non-linear embedding spaces. As described in the previous sections there are at least three possible ways to determine the trait vectors. I found that the traits computed using Sup-

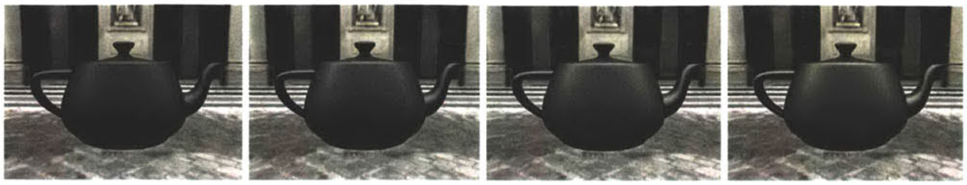

Figure 5-1: *A trait defined using mean difference. The* gold-like *trait is added to BlackOxidizedSteel BRDF. While these results are reasonable, in general the trait vectors found by SVM are judged to be more robust and SVMs were used for the rest of the analysis.*

port Vector Machines are the most robust. The expressions for two other methods (mean difference and Fisher's linear discriminant) are very similar – equations 5.1 and 5.9 differ only in the term S_W^{-1}. Moreover, it is difficult to estimate the matrix S_W well using a small number of samples. I show an example of the trait computed using mean difference and its application to the black-oxidized steel BRDF (Figure 5-1). In the remainder of this chapter the traits are computed using Support Vector Machines.

The results from this trait-based analysis are shown as projections onto the derived trait vectors in Figures 5-2, 5-3, and 5-4. These projections are computed in the linear embedding space given by the non-linear model. Observe that the metallic and specular characteristics are weakly correlated, the specular and diffuse traits are weakly inverse-correlated, and the glossy and diffuse traits are inverse-correlated. This is what one would expect. Note that I do not make attempts to model independent traits in either the trait selection or trait vector derivations. Therefore, it is expected that the addition of a particular trait to an existing BRDF may effect other traits. This lack of parameter independence is a tradeoff that I accept in order to establish perceptually meaningful parameters in my modelling approach. Despite the fact that the parameterization vectors are not orthogonal, they did span the whole 15D non-linear embedding space and, thus, provide an intuitive set of "dials" for users to design materials.

Once trait vectors are established, the user can add and subtract them from data points in the embedding space. In Figure 5-5 I demonstrate four examples of varying user-specified traits using the linear model. The first row shows a teapot rendered using the *GoldPaint* BRDF on the far left, and the effect of adding the *redness* trait in successive steps to the

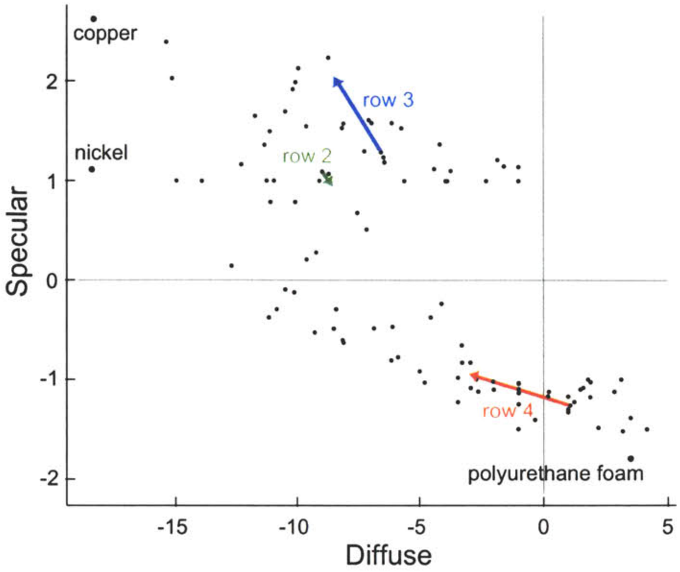

Figure 5-2: Diffuseness *trait vs* specularness *trait. Observe that the* diffuseness *and* specularness *traits exhibit a weak inverse correlation. The green, blue, and red vectors denote projections of the BRDF interpolations shown in the second, third, and fourth rows of Figure 5-6 respectively.*

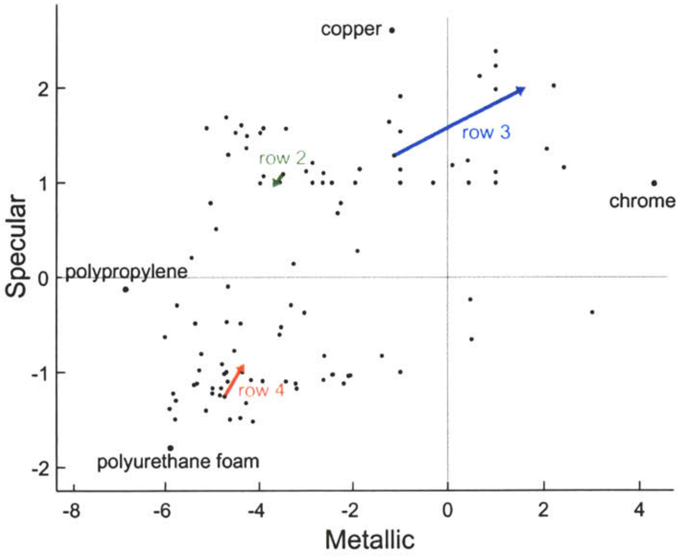

Figure 5-3: Metallic-like *trait vs* specularness *trait. Observe that the* metallic-like *and* specularness *traits exhibit a weak correlation. The green, blue, and red vectors denote projections of the BRDF interpolations shown in the second, third, and fourth rows of Figure 5-6 respectively.*

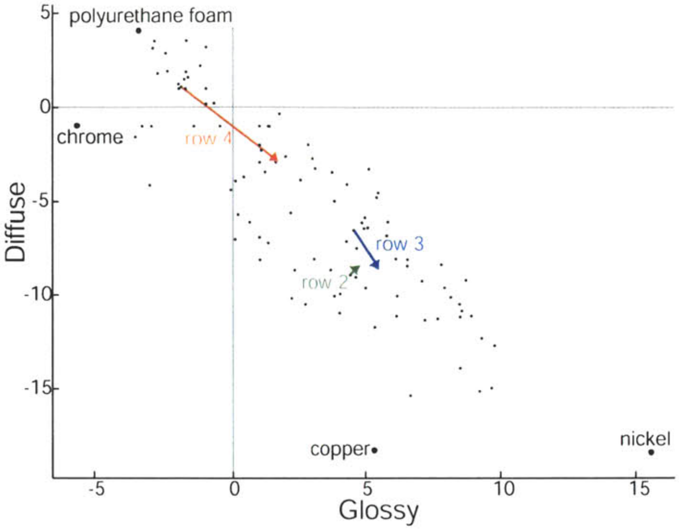

Figure 5-4: Glossiness *trait vs* diffuseness *trait. Observe that the* glossiness *and* diffuseness *traits exhibit an inverse correlation. The green, blue, and red vectors denote projections of the BRDF interpolations shown in the second, third, and fourth rows of Figure 5-6 respectively.*

right. The second row starts from the *SpecularGold* BRDF (left) with successive additions of the *silverness* trait. The third row adds the *gold-like* trait to the *BlueGlossyPaint* BRDF (left). Finally, the fourth row shows the addition of the *specularness* trait to the *BlackMattePlastic* BRDF. It is my experience that the linear model gives reasonable BRDFs if only small displacements are permitted. If the displacement is too large, physically invalid BRDFs result (as illustrated in Figure 4-7).

Next I applied the same trait classifications and Support Vector Machine calculations to the embedding space of the non-linear model. Figure 5-6 demonstrates four examples using this approach. The first row of Figure 5-6 shows the *Copper* BRDF on the left, with successive additions of the *roughness* trait. The second row begins with the *GreenAcrylic* BRDF and shows the addition of the *blueness* trait. The trajectory of this path is also illustrated in Figures 5-2, 5-3, and 5-4. Notice that color-change specification is not particularly correlated with any of the traits used for these projections. Thus, I would expect relatively small movements and no preferred direction. The third row, on the other hand, represents the addition of the *metallic* trait to the *VioletAcryllic* BRDF, whose path is also illustrated in the

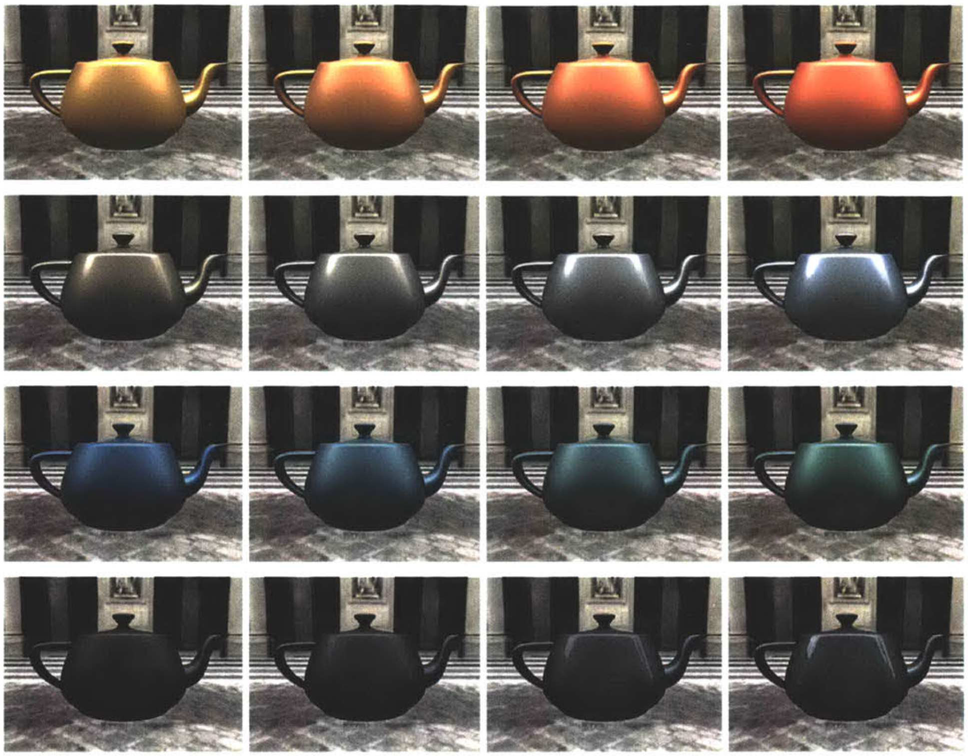

Figure 5-5: *Navigation in the linear space. Each row corresponds to changing one parameter of the model. The first row shows an increase in the* redness *trait applied to the* GoldPaint *BRDF. The second row illustrates an increase in the* silverness *trait applied to the* SpecularGold *BRDF. The third row applies the* gold-like *trait to the* BlueGlossyPaint *BRDF. The fourth row shows an increase in the* specularness *trait applied to the* BlackMattePlastic *BRDF.*

projections in Figures 5-2, 5-3, and 5-4. The path trajectory of this example conforms to my expectations, and its magnitude is large in these visualizations since the *metallic* trait is correlated to the *glossiness* and *specularness* traits used as axes. The fourth row starts with *YellowDiffusePaint* BRDF and shows the addition of the *glossiness* trait, which is depicted as the red path in Figures 5-2, 5-3, and 5-4. The direction of this path is as one would expect, and it has a large magnitude due to the fact the *YellowDiffusePaint* BRDF is located far away from the glossy examples in the projections shown.

Overall, the non-linear basis set results in a more robust model than the linear basis set,

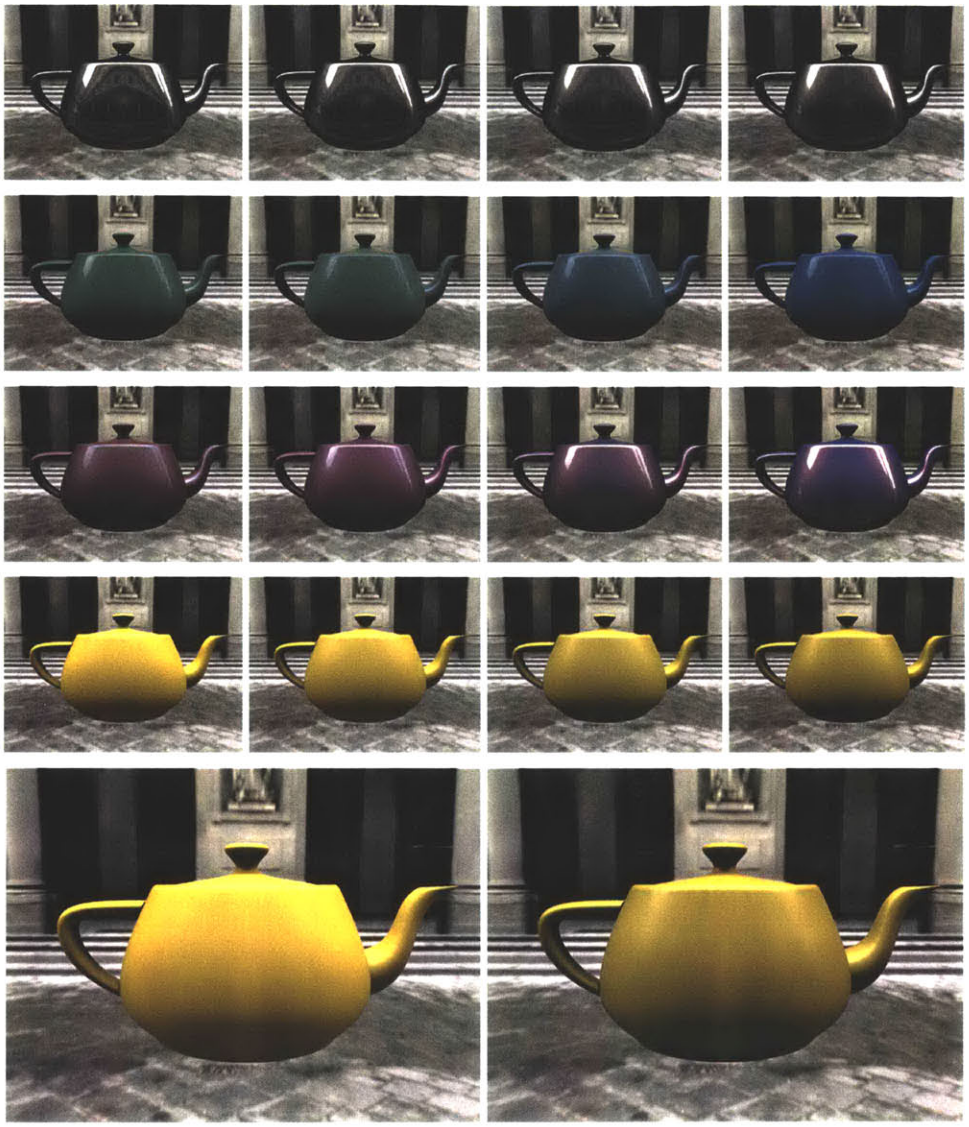

Figure 5-6: *Navigation on the non-linear manifold. Each row corresponds to changing one parameter of the model. The first row shows an increase in the* roughness *trait applied to the* Copper *BRDF. The second row illustrates an increase in the* blueness *trait applied to the* GreenAcrylic *BRDF. The third row applies the* metallic-like *trait to the* VioletAcryllic *BRDF. The fourth and fifth rows show an increase in the* glossiness *trait applied to the* YellowDiffusePaint *BRDF. (The images in the fifth row are the enlarged versions of the first and last image in the fourth row.)*

in that I was able to move large distances within the non-linear embedding space and still generate physically plausible BRDFs with the expected appearance.

5.4 Representing Physical Processes

The modelling approach also allows a user to associate approximate trait vectors with physical processes. This can be done in one of two ways: by fitting a least-squares line to a path of specified BRDFs in the embedding space, or by computing local piecewise difference vectors between examples. As an example, I have modelled a process of metal oxidation. I measured the reflectance changes as a metal was exposed to an acidic environment. It changed from highly specular polished material to black matte material. The acquired four BRDFs determine a path in the embedding space. The intermediate stages are interpolated in the embedding space and backprojected to the sample space (Figure 5-7). Figure 5-8 illustrates another process – rust formation. I used a spatially varying texture to select *rust* levels for each point on the teapot.

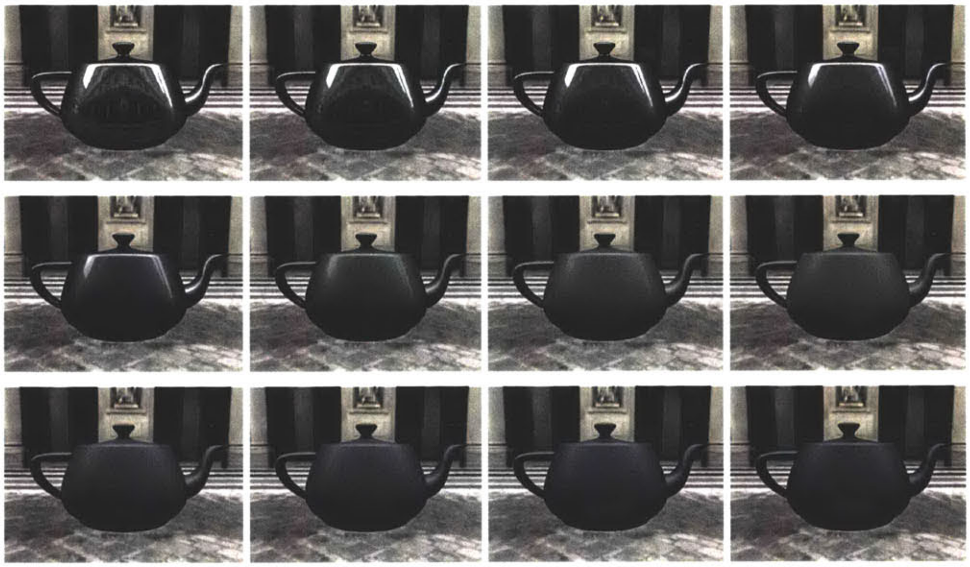

Figure 5-7: *Progression of the steel oxidation process. The data-driven BRDF model also supports interpolation along a physically meaningful path. In this example I start with a completely polished steel sample (upper left) and progressively oxidize it. The final black-oxidized sample is shown on the lower right.*

Figure 5-8: *Rust formation. A spatially varying texture is used to interpolate between reflectance models for each point on the teapot.*

5.5 Discussion

In this chapter I have presented a modelling approach that greatly simplifies navigation on the reflectance manifold. It provides users with a method to specify intuitive reflectance parameters for the data-driven model. However, there are still some unresolved issues that could be addressed. First, in the model derived from non-linear analysis, the embedding space is stretched – the distances on the manifold are not preserved in the embedding space. Therefore, navigation in different places of the embedding space has a different rate of change for the traits. Second, the derived trait vectors are not perceptually uniform. However, this normalization could be done using a method similar to the one described by Pellacini et al. [40]. Third, the fact that the trait vectors are not orthogonal might introduce some problems. It would be desirable to derive an intuitive set of orthogonal trait vectors. Finally, a lot more work could be done to create a good-quality user interface for both trait specification and navigation on the manifold.

Chapter 6

Efficient Storage and Measurement

The main subject of this chapter is efficient storage and measurement of BRDFs. In fact, the solutions for compact BRDF representations imply more efficient measurement (i.e., sampling) procedures for isotropic BRDFs. These procedures can, in principle, significantly accelerate the BRDF acquisition process described in chapter 3. It has previously been observed that wavelets are a very compact representation for BRDFs. Therefore, I start by using a wavelet analysis to compute a single wavelet basis for all BRDFs in my dataset. The coefficients of this basis are a compact representation for most BRDFs. The wavelet analysis also results in an efficient measurement procedure that derives a non-uniform sampling of the BRDF domain required to reconstruct most BRDFs well. Furthermore, I have shown in chapter 4 that each BRDF in the dataset can be represented well using 45 principal components. Therefore, one needs only 45 floating point numbers to represent a BRDF – this is a compact representation. This observation results in an another efficient measurement procedure. The procedure expresses new BRDFs as a linear combination of principal components or source BRDFs. It estimates the coefficients of the linear combination using few (less than a thousand) BRDF samples.

6.1 Wavelets and Discrete Wavelet Transform

First, I give a brief summary of wavelets and wavelet analysis. More details can be found for example in Stollnitz et al. [46] or Strang and Nguyen [47]. Wavelet analysis decomposes a signal into a linear combination of basis functions. These basis functions are shifted and scaled versions of one signal called the mother wavelet and denoted with ψ. The mother wavelet has an effectively compact support – most, if not all, of its energy concentrates in a small region of its domain. Furthermore, the average value of a wavelet is zero. Since the average value of a signal that is decomposed is not zero, one more type of basis function is required. This is the scaling function and is denoted with ϕ. The expression for all wavelet functions derived from the mother wavelet $\psi(x)$ is given by:

$$\psi_m^l(x) = 2^{l/2}\psi(2^l x - m), \tag{6.1}$$

where l is the scale level and m is the shift. Similarly, the scaling functions are defined in terms of the base scaling function:

$$\phi_m^l(x) = 2^{l/2}\phi(2^l x - m). \tag{6.2}$$

The discrete wavelet transform [1] is a fast way of computing the coefficients of wavelet functions for some input signal specified at discrete intervals. The 1D transform proceeds in the number of decomposition steps. The first step applies a set of filters to the input signal. Next, at each successive step the filters are applied to the result of the previous step. The transform extends easily to the multivariate data. (The isotropic BRDFs can be treated as a 3D signal sampled on a discrete grid.) In fact, there are two ways to perform the decomposition: standard and nonstandard [1]. I describe only the nonstandard decomposition since it offers a significant advantage – it allows for a faster evaluation at a point. In the nonstandard decomposition the execution alternates between the variables in the following way: one step of 1D transforms is performed along the axis of the first variable. Then one step of 1D transforms is performed along the axis of the second variable and next along the axis of the third variable. The decomposition is performed in this way for all levels.

When the nonstandard decomposition of the 3D signal is performed the signal is decomposed into a linear combination of 3D basis functions. The scaling function is defined only for level zero and no offsets:

$$\Gamma^0(x,y,z) = \phi\phi\phi(x,y,z) = \phi_0^0(x)\phi_0^0(y)\phi_0^0(z). \tag{6.3}$$

For a given scale level l there are seven different types of 3D wavelet basis:

$$\begin{aligned}
\Gamma^1_{ijkl}(x,y,z) &= \phi\phi\psi^l_{ijk}(x,y,z) &&= \phi^l_i(x)\phi^l_j(y)\psi^l_k(z) \\
\Gamma^2_{ijkl}(x,y,z) &= \phi\psi\phi^l_{ijk}(x,y,z) &&= \phi(x)^l_i\psi(y)^l_j\phi^l_k(z) \\
\Gamma^3_{ijkl}(x,y,z) &= \phi\psi\psi^l_{ijk}(x,y,z) &&= \phi(x)^l_i\psi(y)^l_j\psi^l_k(z) \\
\Gamma^4_{ijkl}(x,y,z) &= \psi\phi\phi^l_{ijk}(x,y,z) &&= \psi(x)^l_i\phi(y)^l_j\phi^l_k(z) \\
\Gamma^5_{ijkl}(x,y,z) &= \psi\phi\psi^l_{ijk}(x,y,z) &&= \psi(x)^l_i\phi(y)^l_j\psi^l_k(z) \\
\Gamma^6_{ijkl}(x,y,z) &= \psi\psi\phi^l_{ijk}(x,y,z) &&= \psi(x)^l_i\psi(y)^l_j\phi^l_k(z) \\
\Gamma^7_{ijkl}(x,y,z) &= \psi\psi\psi^l_{ijk}(x,y,z) &&= \psi(x)^l_i\psi(y)^l_j\psi^l_k(z),
\end{aligned} \tag{6.4}$$

where i, j, and k denote offsets. The 3D input signal $F(x,y,z)$ is expressed as a linear combination of the basis functions Γ:

$$F(x,y,z) = \sum_i \sum_j \sum_k \sum_l \sum_m f_{ijklm} \Gamma^m_{ijkl}(x,y,z). \tag{6.5}$$

A signal $F(x,y,z)$ can usually be compressed well by removing coefficients f_{ijklm} that are near zero. Furthermore, the representation of $F(x,y,z)$ with some of the coefficients removed can be efficiently stored in the wavelet coefficient tree data structure [26]. Each node of the tree stores the coefficients of seven basis functions Γ (the root also stores the coefficient of the scaling function) and the links to eight children. The main advantage of wavelet coefficient tree data structure is that it allows us to evaluate function $F(x,y,z)$ at a point efficiently by pruning away the tree branches that evaluate to zero. Lalonde notes that when using a tree with nonstandard decomposition coefficients one needs to examine only $O(w^3 \log_2 n)$ terms to evaluate $F(x,y,z)$ at a point. (W is the support of the basis function and n is the resolution of data in 1D.)

6.2 Wavelet Representation of BRDFs

Wavelets have been used before to represent BRDFs. Schröder and Sweldens [45] use spherical wavelets to represent 2D slices of a 4D reflectance function. They can represent a slice of the BRDF with several hundreds of coefficients. (The rest of the coefficients are set to zero.) Lalonde [26] extended this work and represents 4D reflectance functions using 4D basis wavelet functions stored in a wavelet coefficient tree. He achieves a very compact representation for a single BRDF.

I propose using a representation similar to the one used by Lalonde. However, there are some important differences. First, my data-driven model represents only isotropic BRDFs (isotropic BRDFs are functions of three variables) while his representation supports general 4D BRDFs. I use 3D nonstandard wavelet decomposition and a corresponding wavelet coefficient tree to encode significant non-zero coefficients as described in the previous section. The second significant difference is data parameterization. Lalonde maps the spherical BRDF data onto a plane using the Nusselt embedding. BRDFs in my data-driven model are parameterized on the sphere using the Rusinkiewicz coordinate frame. Third, the BRDFs he examined were sampled at relatively low resolution (32 disretization bins for each of the four variables). Finally, I use a wavelet coefficient tree to analyze and represent the entire collection of measured BRDFs, not just one BRDF.

Next, I provide the details of wavelet analysis and representation used for the data-driven reflectance model. As shown in chapter 3, each color component (red, green, blue) of a tabulated BRDF is stored as a function of three variables θ_h, θ_d, and ϕ_d. It is represented as a $90 \times 90 \times 180$ three-dimensional array of floating point values. Standard packages that perform discrete wavelet transform require data dimensions that are powers of two. Therefore, I insert each BRDF component into a $256 \times 256 \times 256$ array and pad the rest of the array with zeros. Next, I perform a nonstandard discrete wavelet transform on each 3D array to obtain an array of $256 \times 256 \times 256$ wavelet coefficients. For each color component of a BRDF, I keep the highest coefficients that are necessary to reconstruct the component with 3% precision. The sets of non-zero coefficients are generally different for each color component of a measured BRDF. They are also different for each BRDF in my dataset. However, there is a large degree of coherence between these sets. When I take the union of the 300 sets (there are 100 BRDFs in my dataset and three sets of coefficients for each BRDF), the size of the union set grows to approximately 69,000 wavelet coefficients – 4.7% of the original data. The union of non-zero coefficients corresponds to a set of wavelet functions. The accuracy of this representation is at least 3% but likely better. I call this set of wavelet functions the *Common Wavelet Basis (CWB)* for isotropic BRDFs. Based on these non-zero coefficients I also build the wavelet coefficient tree data structure, which I call the *Common Wavelet Coefficient Tree (CWCT)*. This tree can represent any BRDF in the original dataset and also any BRDF generated by the data-driven model. The main advantage of the common tree is that the data structure needs to be stored only once for all BRDFs. Furthermore, the point evaluation of a BRDF generated using the data-driven model requires only one tree traversal. In the next section I show that the wavelet analysis and the Common Wavelet Basis lead to an efficient measurement procedure for isotropic BRDFs.

6.3 Efficient BRDF Sampling Based on Wavelet Analysis

The measurement procedure I propose in this section focuses on how to efficiently sample isotropic BRDFs. First, I observe that typical BRDFs exhibit high frequencies in only very specific regions of their parameter space (e.g., near specular peaks). Only these regions require dense sampling. Otherwise, BRDFs are smooth and slowly varying over most of their domain, and, thus, require fewer samples for accurate reconstruction in these regions. Non-uniform sampling can be used to exploit this "spatially varying" localized spectrum property, which is characteristic of BRDFs. The precise densities and patterns of this non-uniform sampling are largely a matter of guesswork, and it is likely that one should err on the side of oversampling the function, as I attempted to do in the acquisition of example BRDFs. However, once given a large set of oversampled representative BRDFs, it is possible to analyze the entire corpus in order to reveal the maximum localized signal frequencies for any point in the domain, which in turn implies the maximum necessary sampling frequency for that point. Assuming that the example BRDFs in my database are representative of the entire space of isotropic BRDFs, I can then sample the BRDF of any new material correctly at a lower non-uniform sampling rate without any a priori knowledge.

Standard Fourier analysis could be used to determine the frequency spectrum of the BRDFs in my database. However, the maximum signal frequency would be very high since the Fourier basis functions span the whole domain of the sample space. Consequently, a

Fourier analysis would suggest a dense and uniform sampling. I use wavelet analysis to obtain the maximum signal frequency for each part of the BRDF domain. The advantage of wavelet analysis is its ability to perform localized analysis of a larger signal because the underlying wavelet basis functions vary in both the spatial and frequency domain [47].

Wavelet analysis represents a particular BRDF function as a linear combination of basis functions of varying scale. At the same time it specifies the signal frequency over each interval of the function domain that is required to represent the function. The signal frequency translates directly to the required sampling frequency for each interval [47]. In general, this sampling frequency is only adequate for one particular BRDF. When computing the Common Wavelet Basis, however, I perform a wavelet analysis for all of the BRDFs in the dataset. This wavelet basis defines the maximum signal frequency over each interval of the function domain needed in order to reconstruct any of the measured BRDFs. Using CWB I derive the sampling density and corresponding sampling points (θ_h, θ_d, ϕ_d) that should be measured in order to sample any arbitrary BRDF correctly. My algorithm also reconstructs a dense BRDF representation from the measured BRDF values at the specified sampling points.

Next, I discuss the details of the algorithm – how to define an efficient BRDF sampling pattern, how to compute the CWB coefficients using the BRDF values at these sampling points, and how to reconstruct the full BRDF from the CWB. First, I note that each BRDF component $f_r(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d})$ can be represented as a weighted sum of the CWB functions $h_i(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d})$ as follows:

$$f_r(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d}) = \sum_{i=1}^{n} h_i(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d}) \times c_i, \tag{6.6}$$

where **c** is a vector of coefficients for the CWB functions that need to be computed. I also note that all of the CWB functions are known (for the sake of simplicity I use Haar wavelets) and they can be evaluated at any point $(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d})$. Each BRDF value $f_r(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d})$ produces one linear constraint on the values of the wavelet coefficients **c**. The approximately 69,000 non-zero coefficients as constraints allow us to compute all wavelet coefficients **c** by solving a system of linear equations.

Next, I need to select the sampling points $(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d})$ that produce the linearly independent equations, and, therefore, allow me to compute the wavelet coefficients. I note that there is no unique set of sampling points; thus, I just select one possible set that leads to linearly independent equations. First, I compute one constraint for each original sampling point for a total of 1,458,000 equations. Many of these are linearly dependent. I determine a set of 69,000 equations that are linearly independent. Given these equations and the corresponding BRDF values **g**, I solve the following system of equations:

$$\mathbf{g} = \mathbf{H}\mathbf{c}. \tag{6.7}$$

Matrix **H** is large ($69,000 \times 69,000$) but typically very sparse – usually there are around 40 non-zero elements in a row. This is a result of the small support of the wavelets at higher levels in the wavelet tree. I use the MATLAB sparse matrix routine to directly perform the operation $\mathbf{H}^T/\mathbf{c}^T$ for each color channel (R, G, and B).

Unfortunately, the coefficients for wavelets at the lowest levels (level 0, 1, and 2) are not estimated robustly. There are 8, 84, and 384 of them, respectively. However, I can estimate these coefficients using a different method. I collapse the sparse 69,000 BRDF values from a $256 \times 256 \times 256$ grid to a $16 \times 16 \times 16$ grid by averaging the values. This grid becomes completely filled and I perform the wavelet transform on this low resolution grid. The coefficients of the low resolution grid approximate well the low level coefficients of the high resolution grid.

In order to reconstruct a BRDF I use the estimated coefficients of the CWB and I set coefficients for the wavelets not in the CWB to zero. Then I perform the inverse wavelet transform to compute the BRDF values at the original sampling grid locations.

6.4 Pull-Push Reconstruction of BRDFs

In the previous section I have shown how to reconstruct a BRDF on a uniform grid using 69,000 BRDF samples at specified locations. In this section, I present an alternative reconstruction method that in practice yields lower reconstruction errors.

Since we are given sparse BRDF samples, reconstructing a BRDF at the dense grid can be viewed as a scattered data interpolation problem. One simple and fast method is the pull-push method [16]. This method relies on the pyramid data structure of a progressively downsampled BRDF. Let f^r denote the BRDF grid at the resolution r (the highest, original, resolution of the data structure is for $r = 0$). In this case $f^r_{i,j,k}$ denotes the value of f^r at position (i, j, k). One also needs to store the corresponding hierarchical data structure of weights denoted with w. The algorithm proceeds in three following steps:

The Initialization Step: During the initialization step each measured BRDF sample is used to set the corresponding grid value at the highest resolution $f^0_{i,j,k}$ and the associated weight $w^0_{i,j,k}$ is set to some predefined confidence value c (e.g., $c = 1$). All other weights at the highest resolution are set to 0.

The Pull Step: The *pull* step is applied hierarchically from the highest to the lowest resolution in the BRDF pyramid. Each lower resolution of the pyramid is obtained from the higher resolution version. The lower resolution BRDF values and weights are computed as follows:

$$\begin{aligned} w^{r+1}_{i,j,k} &= \sum_x \sum_y \sum_z \tilde{h}_{x-2i,y-2j,z-2k} w^r_{x,y,z}, \\ f^{r+1}_{i,j,k} &= \frac{1}{w^{r+1}_{i,j,k}} \sum_x \sum_y \sum_z \tilde{h}_{x-2i,y-2j,z-2k} w^r_{x,y,z} f^r_{x,y,z}, \end{aligned} \tag{6.8}$$

where $\tilde{h}$ is a discrete low pass filter – a "tent" sequence $[-1..1] \times [-1..1] \times [-1..1]$ defined as:

$$\tilde{h} = \begin{bmatrix} \frac{1}{64} & \frac{1}{32} & \frac{1}{64} \\ \frac{1}{32} & \frac{1}{16} & \frac{1}{32} \\ \frac{1}{64} & \frac{1}{32} & \frac{1}{64} \end{bmatrix}, \begin{bmatrix} \frac{1}{32} & \frac{1}{16} & \frac{1}{32} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{32} & \frac{1}{16} & \frac{1}{32} \end{bmatrix}, \begin{bmatrix} \frac{1}{64} & \frac{1}{32} & \frac{1}{64} \\ \frac{1}{32} & \frac{1}{16} & \frac{1}{32} \\ \frac{1}{64} & \frac{1}{32} & \frac{1}{64} \end{bmatrix}. \tag{6.9}$$

The Push Step: The *push* step is also applied hierarchically to the BRDF and weight pyramid. It starts at the lowest resolution and progresses to the highest resolution. Low resolution data computed during the *pull* phase is used to fill in the gaps at the higher resolution. If the higher resolution BRDF value has high enough confidence (its weight is larger than c) then the lower resolution value is not used. Otherwise, the higher and lower resolution values are blended together. In order to blend the values, the *push* step first computes temporary values $tw^r_{i,j,k}$ and $tf^r_{i,j,k}$:

$$\begin{aligned} tw^r_{i,j,k} &= \sum_x \sum_y \sum_z h_{i-2x,j-2y,k-2z}, \\ tf^r_{i,j,k} &= \sum_x \sum_y \sum_z h_{i-2x,j-2y,k-2z} f^{r+1}_{x,y,z}, \end{aligned} \tag{6.10}$$

where h is defined on $[-1..1] \times [-1..1] \times [-1..1]$ and expressed as:

$$h = \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}, \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}, \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}. \tag{6.11}$$

Next, the temporary values are blended with the values that are already present at level r according to the formula:

$$\begin{aligned} f^r_{i,j,k} &= tf^r_{i,j,k}(1 - w^r_{i,j,k}) + f^r_{i,j,k} w^r_{i,j,k}, \\ w^r_{i,j,k} &= tw^r_{i,j,k}(1 - w^r_{i,j,k}) + w^r_{i,j,k}. \end{aligned} \tag{6.12}$$

The results of the reconstruction for a few different materials and comparison of the methods are presented in section 6.6.

6.5 Linear Combinations of BRDFs

In chapter 3 I have performed Principal Component Analysis (PCA) over a set of more than 100 densely measured BRDFs. I have shown that each of the measured BRDFs can be represented well by a linear combination of 45 principal components. In this section I show that new BRDFs can be represented equally well using the BRDFs in the original set. It follows that one needs only to estimate appropriate weighting factors for each of the original BRDFs in order to estimate any new BRDF. Since there are only 100 of these coefficients, the number of BRDF samples needed for this estimation should be relatively small.

I represent each densely sampled BRDF as a high-dimensional vector composed of all values for R, G, and B. Let $\mathbf{Y}$ be the matrix of all BRDFs in the original set, $\mathbf{c}$ the vector of coefficients for the linear combinations, and $\mathbf{b}$ the new BRDF we measure. It follows that:

$$\mathbf{Yc} \approx \mathbf{b}. \tag{6.13}$$

This system of equations is overconstrained since it has $90 \times 90 \times 180 \times 3 = 4,374,000$ equations and only 100 unknowns. However, a lot of these equations are linearly dependent. Therefore, I need to select a only small set of equations that allows me to robustly estimate the coefficients $\mathbf{c}$.

I present a strategy for selecting equations that allow for computing the coefficients of the linear combination. Let $\mathbf{X}$ be a matrix composed of some subset of rows of the matrix $\mathbf{Y}$. (As noted before, $\mathbf{Y}$ is a matrix composed of column BRDF vectors.) A good measure of how robustly I can estimate $\mathbf{c}$ is the ratio between the highest and lowest eigenvalue of the matrix $\mathbf{X}^T\mathbf{X}$. The system is well conditioned if this ratio is small. Since finding an optimal set exhaustively is prohibitive, I resort to a simple greedy strategy. I start with an initial set of n constraints. I select a constraint outside of the set $\mathbf{X}$ and one constraint in the set $\mathbf{X}$. I swap them only if the ratio of the eigenvalues decreases in matrix $\mathbf{X}^T\mathbf{X}$ with the constraints swapped. I repeat this procedure until $\mathbf{X}$ converges to a stable set and I perform this procedure for different set sizes n.

This procedure guarantees that the system is numerically well conditioned, which in turn makes it robust to perturbations of the constraints. It also approximates a "most informative set" of measurements. In order to expect a good generalization of the known BRDFs, I must also ensure that the system is well overconstrained. I found that using $n = 800$ linear constraints provides a robust computation of the coefficients, while adding more constraints does not improve the solution. This implies that I can measure new BRDFs using only 800 samples. (At each point I measure either R, G, or B.) I note that each of the 800 equations corresponds to a specific value of $(\theta_\mathrm{h}, \theta_\mathrm{d}, \phi_\mathrm{d})$ and does not depend on the value of the BRDF at that point.

The procedure presented here is well defined, simple, and fast. In order to compute the coefficients $\mathbf{c}$ I need only to compute a pseudo-inverse of the 800×100 matrix and perform one vector-matrix multiplication. In contrast to fitting samples to analytic BRDF models, this procedure is not dependent on a good initial guess.

6.6 Reconstruction Results

To validate my methods, I densely measured four additional isotropic BRDFs: dark-red paint, gold paint, orange plastic, and aluminum-bronze. These materials are substantially different from any of the materials in the original collection.

First, I show that these BRDFs can be represented well using only the coefficients in the CWB (which of course has been computed without these materials). Figure 6-1 compares the original BRDFs and the BRDFs expressed with the CWB for different angles of incident illumination. The errors for each of the BRDFs are: dark-red paint: 0.7%, gold paint: 0.9%, orange plastic: 2.1%, and aluminum-bronze - 1.2%. We conclude that our common wavelet basis represents these new BRDFs well.

Next, I reconstruct these BRDFs from 69,000 samples specified by the linear constraints. The results of this reconstruction are shown in Figure 6-2. The errors for each of the reconstructed BRDFs compared to the original BRDFs are: dark-red paint: 1.0%, gold paint: 1.3%, orange plastic: 3.2%, and aluminum-bronze: 1.2%. Although the errors are relatively small, we observe some ringing artifacts that are typical for the non-smooth Haar wavelets. Smooth wavelets should yield better looking results.

I also reconstruct the same BRDFs using the pull-push algorithm. I use the same 69,000 sparse BRDF samples. This solution yields better results. The errors for each of the pull-push reconstructed BRDFs compared to the original BRDFs are: dark-red paint: 0.6%, gold paint: 0.9%, orange plastic: 2.5%, and aluminum-bronze: 1.1%. The results of the pull-push reconstructions are shown in Figure 6-3.

Next I show that the reflectance of these materials can be represented well using a linear combination of 100 BRDFs from the original collection (which also did not contain these BRDFs). In Figure 6-4 I show the comparison between the original densely sampled BRDFs and the corresponding reconstructed BRDFs using just 800 BRDF samples. The errors for each of the reconstructed BRDFs as a linear combination of BRDFs compared to the original BRDFs are: dark-red paint: 1.8%, gold paint: 1.8%, orange plastic: 4.3%, and aluminum-bronze: 2.5%.

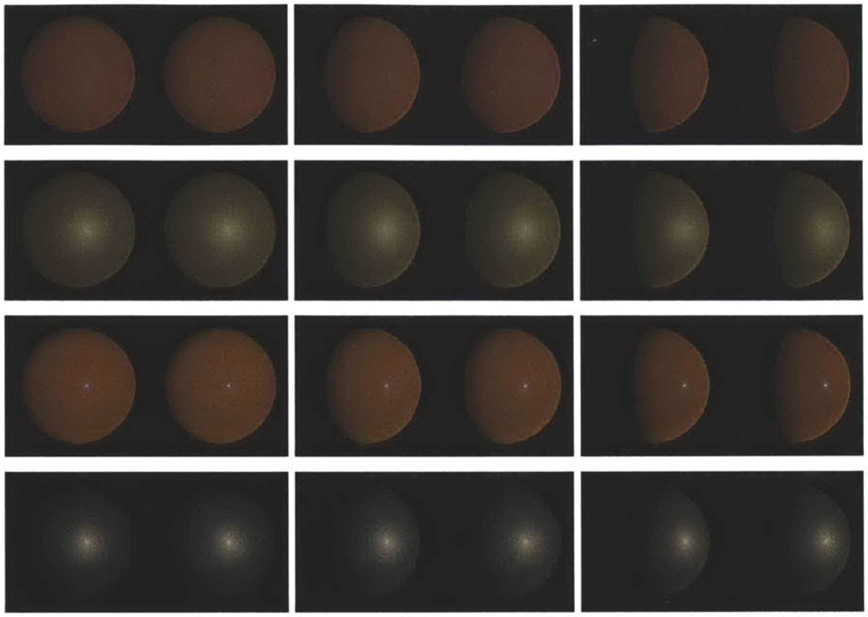

Figure 6-1: *Comparison of BRDFs expressed in common wavelet basis (left) with the original densely sampled BRDFs (right). Each row shows a different BRDF (first row: dark-red paint, second row: gold paint, third row: orange plastic, fourth row: aluminum-bronze).*

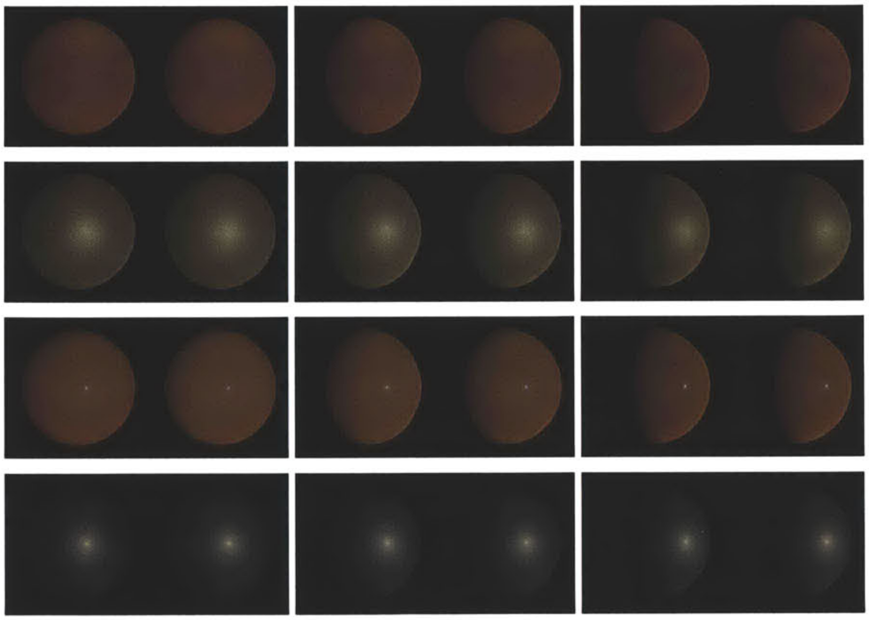

Figure 6-2: *Comparison of wavelet reconstructed BRDFs using 69,000 sparse samples (left) with the original densely sampled BRDFs (right). (First row: dark-red paint, second row: gold paint, third row: orange plastic, fourth row: aluminum-bronze.)*

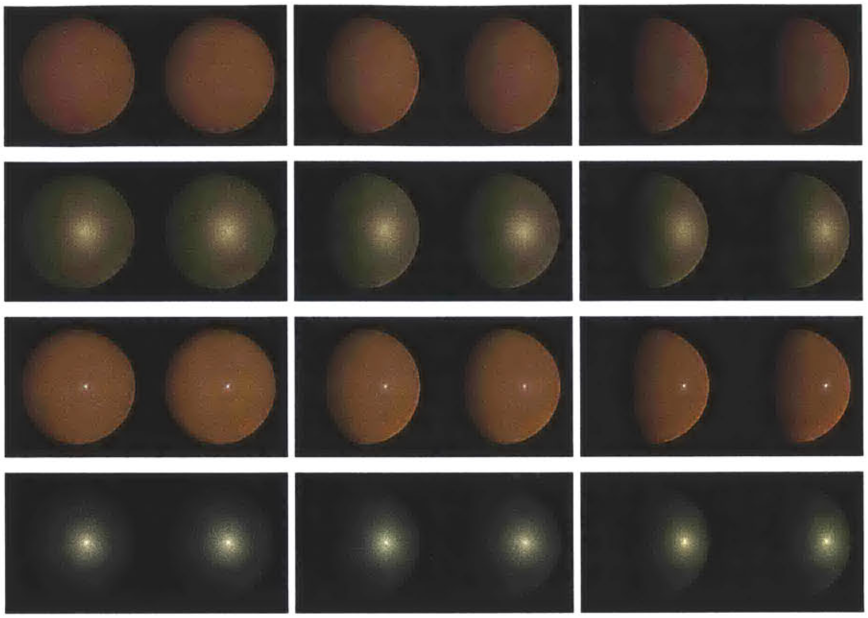

Figure 6-3: *Comparison of pull-push reconstructed BRDFs using 69,000 sparse samples (left) with the original densely sampled BRDFs (right). (First row: dark-red paint, second row: gold paint, third row: orange plastic, fourth row: aluminum-bronze.)*

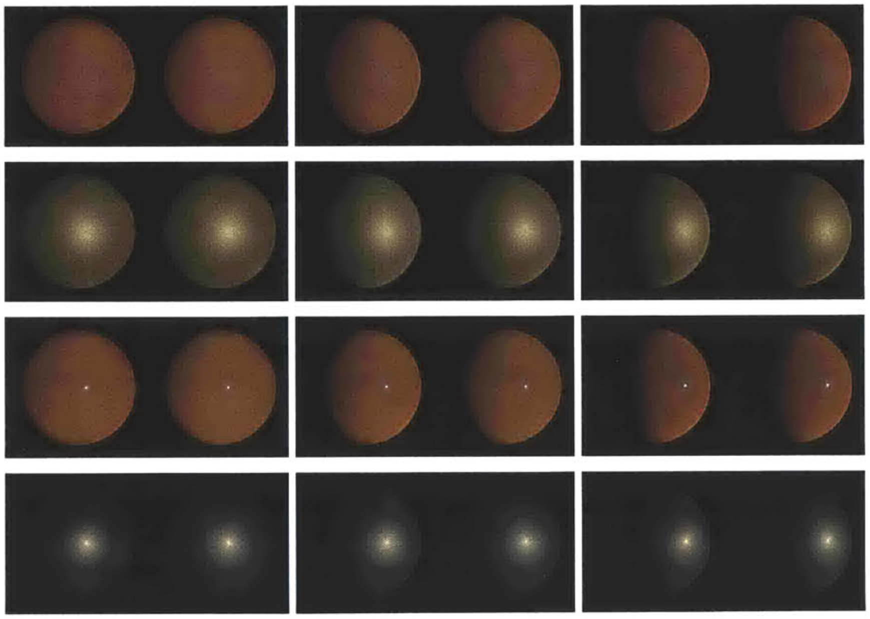

Figure 6-4: *Comparison of BRDFs reconstructed as linear combinations of original BRDFs using 800 samples (left) with the original densely sampled BRDFs (right). Each row shows a different BRDF (first row: dark-red paint, second row: gold paint, third row: orange plastic, fourth row: aluminum-bronze).*

6.7 Summary

In this chapter I have described both efficient BRDF representations and efficient BRDF sampling procedures. I have also shown that they are inherently related. The described sampling procedures require orders of magnitude fewer samples than standard BRDF measurement procedures. This implies that BRDF measurement can be done efficiently using gonio-reflectometers that measure planar material samples. This, in turn, allows capture of a much wider selection of different materials.

All presented methods have their advantages and disadvantages. The methods based on wavelet analysis (CWB reconstruction and pull-push reconstruction) require 69,000 measurements. However, these methods are independent of any BRDF database and can be applied immediately. The linear combination of BRDFs requires only 800 measurements, but it relies on the availability of the BRDF database.

Chapter 7

Conclusions and Future Work

This chapter summarizes the contributions of my dissertation. It outlines my data-driven BRDF modelling approach, the data analysis applied to this model, and presented algorithms. Furthermore, it provides many new directions for future work and suggests how to extend the usefulness of the data-driven reflectance model.

7.1 Conclusions

In this dissertation I have introduced a new approach for modelling isotropic BRDFs. My model generates new surface reflectance models by forming combinations from a set of densely sampled, acquired BRDFs. I am hopeful that data-driven reflectance modelling approaches, like this, can greatly expand the range of material models used in computer graphics rendering.

In order to develop an effective and efficient interpolation scheme, I chose to first analyze the inherent dimensionality of my dataset. I have used both linear (subspace) and non-linear manifold analysis to model a large sample of acquired real-world BRDFs. The results of this analysis are suggestive of the overall structure of isotropic BRDFs. Specifically, I found that the linear subspace model lent itself to the creation of physically implausible BRDFs, and a large number of dimensions (around 45) were required to adequately represent my measurements. Nonetheless, I still found the linear subspace model to be useful for interpolation over small distances. The non-linear model, on the other hand, was much more compact in its dimensionality (around 15 dimensions for the same accuracy as the 45-dimension linear subspace model), and more robust in its ability to interpolate plausible BRDFs over long distances. However, I caution against over-generalizing from my results. I am comfortable in saying that my modelling approach effectively represents my dataset, but the sample size is still relatively small to draw conclusions regarding the fundamental nature of isotropic BRDFs. However, I am optimistic that techniques like mine can be used to greatly expand our knowledge in these areas.

I also have demonstrated methods for defining intuitive parameters for navigating within BRDF models. These techniques can easily be customized for a range of industrial and artistic applications. Furthermore, they can be personalized for individual use or made objective by incorporating physical measurements.

The advantages of my data-driven BRDF model include a high degree of realism, a perceptually meaningful parameterization, relative ease of modelling for complex surface materials, and speed of evaluation. The main disadvantage of the model is its size. I believe that the model I propose can easily be incorporated into existing rendering systems.

In this dissertation I have also presented two novel approaches for measuring isotropic BRDFs. These procedures significantly reduce the number of required measurement samples. My approaches are based on the analysis of densely sampled BRDFs of a large collection of different materials. The first approach analyzes the general surface reflectance function to determine the local signal variation at each point in the function's domain. Then, wavelet analysis is used to derive a common wavelet basis for all acquired BRDFs and the corresponding non-uniform sampling pattern. I have also shown how to reconstruct a BRDF on a uniform grid from these non-uniform samples. In a second approach I showed that isotropic BRDF of any material can be expressed as a linear combination of other surface reflectance functions. Furthermore, my analysis provides a reduced set of sampling points that allows one to robustly estimate the coefficients of this linear combination. These procedures can dramatically shorten the acquisition time for isotropic reflectance measurements.

7.2 Extensions and Future Work

I believe that the work presented in this dissertation will stimulate more research in many different directions. In this section, I will describe the most important areas for future work.

7.2.1 Analysis of Other Surface Reflectance Functions

The most obvious extension of this dissertation would be to apply the subspace and nonlinear analysis to more complex surface reflectance functions. The main difficulty of this research is obtaining densely sampled data.

The first logical step would be to perform the same analysis on full 4D BRFDs. These BRDFs are capable of representing anisotropy. In this case the acquisition and storage of these BRDF become even more difficult. Note that the acquisition could not be performed using the method shown in this dissertation. One would have to use mirror/lens systems to perform fast acquisition. (Measurement using a gonio-spectro-reflectometer is not practical.) However, the number of required images would still be extremely large. For example, if a spacing similar to the one in this dissertation is used for isotropic measurements, I have 90x90x360x180x3 = 1,574,640,000 samples. If I use four bytes for each BRDF sample, then I need 6,298,560,000 bytes (more than 6GB) to store one BRDF. Even if I use the whole imager to acquire a 90x360x3 slice of the function, I would still need to take more than 16,000 pictures to acquire one BRDF. A better strategy might be to use an isotropic data-driven model developed in this dissertation and determine its parameters for each value of ϕ_h. This would significantly reduce the number of measurements.

The next logical step is to extend the data-driven model to support macro-scale surface variations typically described by bidirectional texture functions (BTFs). BTFs are 6D functions that describe surface reflectance functions over real-world surfaces. BTFs have not been measured densely before. Usually only sparse set of measurements is performed (200 per each texon) [12]. Then the reflectance of each texon is fit to an analytic model. Dense measurements of BTFs would require a lot of measurements and storage. For example, if I use a surface patch with 64x64 texels, then the size of one BTF is 24.5 TB.

Measuring bidirectional surface scattering distribution function (BSSRDF) [37] is even harder. This function assumes that the light can penetrate into the surface of the material – it does not assume the light exits the surface at precisely the same location as it enters the surface. In other words, BSSRDF describes the light transport between any incoming and outgoing rays over a surface patch. Thus, BSSRDF is a function of eight variables. Dense measurements of this function have not been performed yet. Some researchers [19] have tried to estimate the parameters of an analytic model.

An alternate direction to building more complex models of reflectance is to build more specialized models. For example, one could narrow the model to encompass only reflectance of human skin or of a class of paints. One would have to measure a reasonable number of BRDFs of different examples from the selected domain and then perform the analysis on this dataset. It is almost certain that the reflectance of a specialized reflectance model forms a much simpler and even lower dimensional manifold than the one for general isotropic reflectance.

Similarly, one should perform the analysis of space of real-world textures before building a data-driven model for BTFs. The space of real-world textures is certain to be a non-linear manifold in a high-dimensional space. This space is not linear because a linear combination of two textures might result in an image that does not look like a real texture. It is also likely that the dimensionality of the texture manifold is high. Therefore, one would need many texture samples to properly represent this manifold.

7.2.2 Real-Time Rendering

Current Graphics Processing Units (GPUs) offer adequate processing power in order to implement a real-time shader for my data-driven BRDF model. The goal would be to render each point in a scene with a potentially different BRDF. There have been some attempts to render tabulated BRDF data in real time using separable approximations [22] or homomorphic factorization [36]. However, these approaches are limited to a few simple BRDFs. Each BRDF has to be stored in a separate data structure. On the other hand there also have been approaches to render spatially varying BRDFs [35]. However, in that case the analytic model, not tabulated data, is used – each texel stores parameters of an analytic model.

Representing spatially varying BRDFs using my model is reasonably compact – one needs to store the coefficients of the principal components. (This results in roughly 45 parameters per texel.) The only difficulty is to compactly represent and efficiently evaluate the principal components. (Each component has the size of one measured BRDF that is 17 MB.) One possible solution is to use a common wavelet tree (similar to the common wavelet basis) to store all principal components efficiently. The evaluation of the principal components from this compressed representation can also be done efficiently. Using the tree traversal to prune the branches not in the support, one needs only to evaluate about 40 terms (compared to 69K wavelet terms required to represent each principal component).

7.2.3 Inverse Methods

I believe that my data-driven reflectance model will be used to solve a number of inverse rendering problems. Inverse rendering attempts to recover material properties and the light transport in a scene from an acquired set of images. Usually the scene geometry is assumed to be known. The reflectance of the scene materials is approximated with analytic BRDF models [57]. My data-driven model is capable of representing reflectance more faithfully than analytic models. Moreover, the estimation process is much simpler since my model is linear and does not require non-linear optimization.

This reflectance model can be also used to improve a number of shape from X techniques. These methods usually use very simple analytic reflectance models. Typically they approximate the reflectance with only a diffuse term.

My data-driven model provides more than just coefficients of linear combination. The points on the manifold are labelled with a corresponding material type. Therefore, by estimating parameters of my model I would automatically obtain the material type that generated that reflectance. This might be very useful for solving problems like segmentation, detection, or recognition. The scene could be segmented into patches of similar materials (metal, paint, skin, etc.) to improve the unsupervised segmentation algorithms. The detection methods could also be improved. In the case of face detection, knowing that some scene regions have human skin reflectance would immediately simplify the problem.

A new avenue for future research would be to provide a way of designing materials virtually while providing the mapping back to the underlying physical mixtures. One would be able to navigate on the reflectance manifold and design a material with the desired reflectance properties. Then, one would get back mixing proportions of base materials that would produce the physical sample of the desired material. To accomplish this, the reflectance manifold would have to be sampled very densely. I believe that the correspondence between mixing reflectance and mixing materials would typically work only for closely spaced samples (small distances on the manifold). This work could have a big impact in the paint industry. It would mean that new paints could be designed and visualized without physically manufacturing them.

Appendix A

Rendering

This appendix describes how to apply the data-driven reflectance model in computer graphics rendering applications. I use Dali (a global illumination renderer) developed by Henrik Wann Jensen to apply the data-driven model to object surfaces and render the pictures. Dali is a ray-tracing renderer that incorporates photon-mapping to model global-illumination effects [18]. I use three different shaders (small programs executed by the renderer when interaction of light with a surface occurs) that utilize the data-driven reflectance model. First, I describe the simplest shader that takes into account direct illumination only. Next, I describe a shader that is used in a Monte Carlo path tracing to obtain a global illumination solution. The third shader uses a compressed representation of the data-driven model (wavelet tree) to evaluate the reflectance function.

In general, there are two ways to use the data-driven reflectance model to render pictures. In the first method, for each material in the scene a separate tabulated BRDF is generated from the source data, and then, we can use these tabulated BRDFs as input to the shader. In the second method, each BRDF is always represented as coefficients of the linear combination of the most significant principal component vectors. (Since I use 45 principal components each BRDF is expressed by 45 floating point numbers.) During rendering, the shader needs to compute the value of the linear combination (45 multiplications and 44 additions) in order to evaluate a BRDF for a given incoming and outgoing light direction. These methods present an obvious trade-off between storage and computation. The first method is very fast since evaluating a BRDF is just a single memory look-up. The disadvantage is that each BRDF is stored separately as a tabulated data structure that requires about 17MB of memory. Using hundreds or thousands of materials in the same scene would be impossible because of memory requirements. When using the second method one needs to store 45 principal components (this takes 765MB) but then each BRDF requires only 45 floating point numbers. This is a very compact representation for an arbitrary BRDF. It practically enables using millions of different materials in one scene – a different BRDF for each point in the scene.

A.1 Simple Direct Illumination

In the case of direct illumination, both the direction towards the light source and the direction towards the viewer are known. Therefore, the shader computes the variables (θ_i,θ_r, ϕ_{diff}) based on these directions and the surface normal. Since the tabulated BRDFs or the basis functions (e.g., principal components) are stored in the Rusinkiewicz coordinate system, the variables (θ_i,θ_r, ϕ_{diff}) are converted to (θ_h, θ_d, ϕ_d). The shader computes the expression for the reflected radiance L_r according to the following formula:

$$L_r(\theta_r,\phi_r) = f_r(\theta_h,\theta_d,\phi_d)E_i(\theta_i,\phi_i) = f_r(\theta_h,\theta_d,\phi_d)L_i(\theta_i,\phi_i)\cos\theta_i, \tag{A.1}$$

where E_i and L_i denote incident irradiance and radiance, respectively. The value of reflected radiance is computed for red, green, and blue components of the light spectrum. Furthermore, if there is more than one light source in the scene, equation A.1 needs to be evaluated separately for each light source; the reflected radiance is a sum of the contributions from all light sources. Some of the pictures rendered using direct illumination only are shown in the Figure A-1.

A.2 Monte Carlo Path Tracing

The shader capable of supporting global-illumination rendering is more complicated than the one for direct illumination. In general, in this scenario the transport of radiant flux proceeds backwards. Therefore, the incident irradiance is not known ahead of time. This

Figure A-1: *Renderings using direct illumination. Aluminum-bronze teapot on a pink-felt table (top) is illuminated using two area light sources. Five teapots (gray marble, nickel, copper, oxidized steel, and blue metallic paint) are rendered under the same two area light sources (bottom).*

means that the direction of the reflected ray is known but not the value of the reflected radiance. In order to estimate the reflected radiance, the shader needs to estimate the incident irradiance from all directions to perform the integration with the BRDF according to the following formula:

$$L_r(\theta_r, \phi_r) = \int_{\Omega} f_r(\theta_h, \theta_d, \phi_d) L_i(\theta_i, \phi_i) \cos\theta_i d\omega_i, \tag{A.2}$$

where Ω defines the incoming light hemisphere (the integration is preformed for the whole hemisphere of incident light directions), and L_r is the radiance in the desired reflected direction. The value of the integral can be approximated by a sum of discrete elements. However, to obtain a reasonable estimate the hemisphere needs to be subdivided into tens of thousands elements and one ray needs to be sent for each of these directions. The deterministic solution is therefore expensive, so in practice a randomized Monte Carlo algorithm is used to form effective estimators.

A.2.1 Basic Monte Carlo Integration

First, I give a short overview of Monte Carlo integration method. (My outline is based on the one presented by Veach [54].) The goal of the Monte Carlo integration is to use random sampling to evaluate the integral:

$$I = \int_{\Omega} f(x) d\mu(x). \tag{A.3}$$

If N independent samples $[X_1, \ldots, X_N]$ of variable x sampled according to some probability density function $p(x)$ are used, then the estimator for the integral I is given by the formula:

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}. \tag{A.4}$$

Furthermore, the variance of the estimator F_N is calculated by:

$$Var[F_N] = Var\left[\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\right] = \frac{1}{N^2} \sum_{i=1}^{N} Var\left[\frac{f(X_i)}{p(X_i)}\right] = \frac{1}{N} Var\left[\frac{f(X)}{p(X)}\right]. \tag{A.5}$$

In the context of rendering, high variance results in noisy images. Therefore, to make the images low-noise one needs to decrease the variance. This can be done by increasing the number of samples N or making density function $p(x)$ close to the integrand $f(x)$.

A.2.2 Sampling Random Variables

Before I give details of shaders based on Monte Carlo integration, I will briefly outline how to generate random samples according to a specified probability density function $p(x)$. This overview also follows the description of Veach [54]. I describe two important techniques: transformation (or inversion) method and rejection sampling.

In transformation method the cumulative density function $P(x)$ of the probability density function $p(x)$ is computed first. Next, the inverse P^{-1} of function $P(x)$ is computed. If the variable y is sampled uniformly from [0,1], then the corresponding variable $P^{-1}(y)$ is sampled from distribution $p(x)$. The transformation method extends to higher dimensions easily – marginal and conditional distributions need to be computed first; next, each dimension is inverted separately.

Rejection sampling is the second important method. In rejection sampling some distribution $q(x)$ (e.g., uniform distribution) and some constant M are defined such that for all x: $p(x) \leq Mq(x)$. Then the following function generates samples according to distribution $p(x)$:

```
function RejectionSampling()
    while (true)
        Sample $X_i$ according to $q$.
        Sample $U_i$ uniformly on $[0,1]$.
        if $U_i \leq p(X_i)/(Mq(X_i))$ then return $X_i$
```

A.2.3 Uniform Sampling

The simplest way of estimating of the reflected radiance using Monte Carlo integration is to assume that nothing is known about the integrand – the product of the BRDF and the incident radiance. Thus, in this case distribution $p(x)$ is assumed to be uniform – the rays are sent in uniformly selected random directions on the hemisphere. The estimator for the reflected radiance is computed according to the formula:

$$L_{\mathrm{r}}(\theta_{\mathrm{r}}, \phi_{\mathrm{r}}) = \frac{1}{N} \sum_{j=1}^{N} f_{\mathrm{r}}(\theta_{\mathrm{r}}, \phi_{\mathrm{r}}, \theta_{\mathrm{i}}^{\mathrm{j}}, \phi_{\mathrm{i}}^{\mathrm{j}}) L_{\mathrm{i}}(\theta_{\mathrm{i}}^{\mathrm{j}}, \phi_{\mathrm{i}}^{\mathrm{j}}) \cos \theta_{\mathrm{i}}^{\mathrm{j}}, \tag{A.6}$$

where N is the number of rays sent. This approach works reasonably well for relatively diffuse BRDF and low-frequency illumination. However, the method produces noisy images for glossy or specular BRDFs or high-frequency lighting even if the number of samples N is large.

A.2.4 Importance Sampling

The basic idea of importance sampling is to generate sampling rays according to density function $p(x)$ that is similar to the integrand $f(x)$. In fact the best choice for $p(x)$ is:

$$p(x) = \frac{f(x)}{\int_{\Omega} f(y) d\mu(y)}, \tag{A.7}$$

which gives zero variance of the estimator. Unfortunately the integrand, the reflected radiance – a product of incident irradiance and BRDF, is unknown. While it is possible to make some complex assumptions about the distribution of the incident radiance, I focus on the effect of the BRDF on the value of the integral. In particular, I assume that incident radiance L_{i} is distributed uniformly and the density function $p(x)$ is influenced by the BRDF and the foreshortening term. In this case the estimator for the reflected radiance L_{r} becomes:

$$L_{\mathrm{r}}(\theta_{\mathrm{r}}, \phi_{\mathrm{r}}) = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{p(\theta_{\mathrm{r}}, \phi_{\mathrm{r}}, \theta_{\mathrm{i}}^{\mathrm{j}}, \phi_{\mathrm{i}}^{\mathrm{j}})} f_{\mathrm{r}}(\theta_{\mathrm{r}}, \phi_{\mathrm{r}}, \theta_{\mathrm{i}}^{\mathrm{j}}, \phi_{\mathrm{i}}^{\mathrm{j}}) L_{\mathrm{i}}(\theta_{\mathrm{i}}^{\mathrm{j}}, \phi_{\mathrm{i}}^{\mathrm{j}}) \cos\theta_{\mathrm{i}}^{\mathrm{j}}, \tag{A.8}$$

where $p(\theta_{\mathrm{r}}, \phi_{\mathrm{r}}, \theta_{\mathrm{i}}^{\mathrm{j}}, \phi_{\mathrm{i}}^{\mathrm{j}})$ denotes the importance function that generates ray sampling.

First, note that each time the shader is invoked the reflected ray direction is known; therefore, the problem can be decomposed into a set of problems of generating sampling according to a 2D probability density function – one for each reflected ray direction $(\theta_{\mathrm{r}}, \phi_{\mathrm{r}})$. Second, my data-driven model describes only isotropic BRDF; thus, the reflected ray direction needs to be described by θ_{r}. Let f_{sum} denote:

$$f_{\mathrm{sum}}(\theta_{\mathrm{r}}, \theta_{\mathrm{i}}, \phi_{\mathrm{diff}}) = f_{\mathrm{r}}^{red}(\theta_{\mathrm{r}}, \theta_{\mathrm{i}}, \phi_{\mathrm{diff}}) + f_{\mathrm{r}}^{green}(\theta_{\mathrm{r}}, \theta_{\mathrm{i}}, \phi_{\mathrm{diff}}) + f_{\mathrm{r}}^{blue}(\theta_{\mathrm{r}}, \theta_{\mathrm{i}}, \phi_{\mathrm{diff}}), \tag{A.9}$$

where f^{red},f^{green},f^{blue} denote the red, green, and blue components of the BRDF. Then the importance function for a particular reflected ray direction θ_{r} is a function of two variables

$(\theta_\mathrm{i}, \phi_\mathrm{diff})$ and is computed as follows:

$$p(\theta_\mathrm{i}, \phi_\mathrm{diff}|\theta_\mathrm{r}) = \frac{f_\mathrm{sum}(\theta_\mathrm{r}, \theta_\mathrm{i}, \phi_\mathrm{diff}) \cos\theta_\mathrm{i}}{\int\int f_\mathrm{sum}(\theta_\mathrm{r}, \theta_\mathrm{i}, \phi_\mathrm{diff}) \cos\theta_\mathrm{i} d\theta_\mathrm{i} d\phi_\mathrm{diff}}. \tag{A.10}$$

I then use the transformation method or rejection sampling to generate rays according to importance distribution $p(\theta_\mathrm{i}, \phi_\mathrm{diff}|\theta_\mathrm{r})$. Since the distribution $p(\theta_\mathrm{i}, \phi_\mathrm{diff}|\theta_\mathrm{r})$ is a function of two variables, I split the sample generation into two steps: the first step determines θ_i; the second step determines ϕ_diff for the previously selected value of θ_i. In the first step I use the following marginal density function:

$$p_{\theta_\mathrm{r}}(\theta_\mathrm{i}) = \frac{\int f_\mathrm{sum}(\theta_\mathrm{r}, \theta_\mathrm{i}, \phi_\mathrm{diff}) \cos\theta_\mathrm{i} d\phi_\mathrm{diff}}{\int\int f_\mathrm{sum}(\theta_\mathrm{r}, \theta_\mathrm{i}, \phi_\mathrm{diff}) \cos\theta_\mathrm{i} d\theta_\mathrm{i} d\phi_\mathrm{diff}}. \tag{A.11}$$

In order to make the sampling process efficient, the value of the marginal density function $f_{\theta_\mathrm{r}}(\theta_\mathrm{i})$ needs to be stored for each θ_r and θ_i.

In the second step there are two alternatives – I can use either the transformation method or rejection sampling. In the transformation method I store 1D conditional cumulative distribution function $P(\phi_\mathrm{diff}|\theta_\mathrm{r}, \theta_\mathrm{i})$ – a function of ϕ_diff for all values of $(\theta_\mathrm{r}, \theta_\mathrm{i})$. I obtain one sampling ray by generating a random value between 0 and 1 and then finding the corresponding value of ϕ_diff using a binary search. This approach has two drawbacks: first, it requires the binary search to find the corresponding value of ϕ_diff; second, it requires storing $P(\phi_\mathrm{diff}|\theta_\mathrm{r}, \theta_\mathrm{i})$ that has comparable storage size to the BRDF. The second alternative uses rejection sampling. It requires storing the value M – the maximum value for each pair $(\theta_\mathrm{r}, \theta_\mathrm{i})$:

$$M(\theta_\mathrm{r}, \theta_\mathrm{i}) = \max_{\phi_\mathrm{diff}} f_\mathrm{sum}(\theta_\mathrm{r}, \theta_\mathrm{i}, \phi_\mathrm{diff}) \cos\theta_\mathrm{i}. \tag{A.12}$$

I generate a bin k for variable ϕ_diff uniformly at random and a number r also uniformly at random between 0 and 1. Then if $f_\mathrm{sum}(\theta_\mathrm{r}, \theta_\mathrm{i}, k)/M(\theta_\mathrm{r}, \theta_\mathrm{i})$ is larger than number r then the direction is selected, otherwise the procedure is repeated. Rejection sampling does not require storing cumulative distribution functions and the binary search. Rejection sampling is fast if M is relatively similar to the rest of the values.

To summarize, the importance sampling allows me to render low-noise pictures using fewer than 1000 rays even when illumination or BRDF has high frequencies. To demonstrate this I use some source BRDFs and high-dynamic range environment maps to simulate complex natural illumination. Some example renderings are shown in Figure A-2.

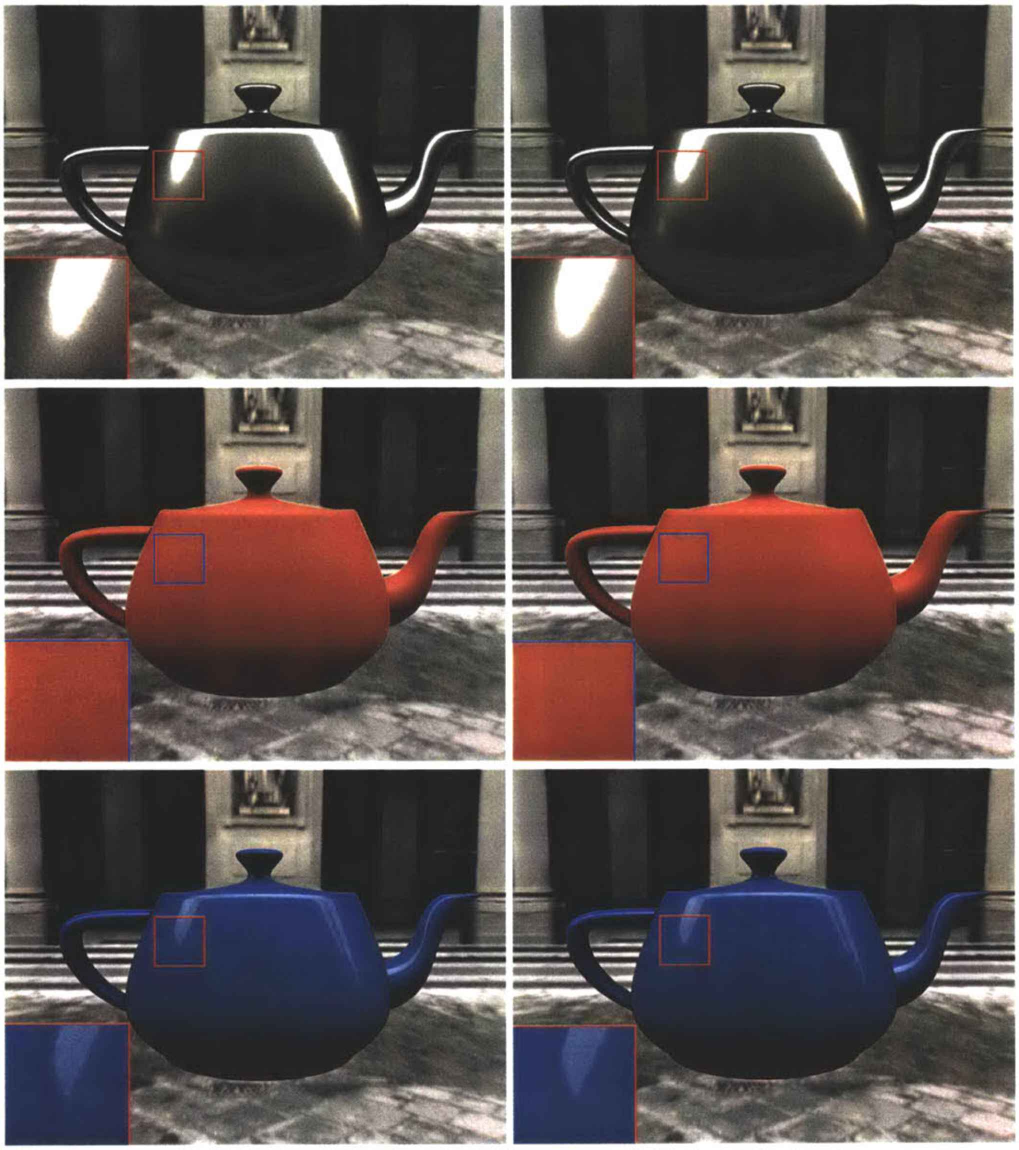

Figure A-2: *Renderings under complex natural illumination. Importance sampling is used to decrease the noise: first column: 100 samples, second column: 1000 samples.*

A.3 Rendering Using Wavelet-Compressed BRDFs

Rendering pictures using tabulated BRDFs or using tabulated principal components of the data-driven model requires a lot of memory. This section discusses how to reduce this memory requirement by using wavelet representation of BRDFs, and how to render pictures directly from this compressed representation.

In general, rendering directly from wavelet-compressed representation trades the speed of evaluation for compact representation. When using tabulated BRDF representation, each BRDF value is readily available. When using wavelet representation, each BRDF value is a linear combination of the basis wavelet functions.

My implementation of the wavelet shader builds upon to the work of Lalonde [26]. His wavelet coefficient tree represents only a single BRDF. I have proposed in chapter 6 a common wavelet basis (or common wavelet coefficient tree) as a compressed basis for the BRDFs in my dataset. I have shown that this basis can be used both for efficient storage of the BRDFs in my dataset and also principal components required to represent these BRDFs. In this section I show how to render pictures directly from this compressed representation.

The procedure for building and traversing the common wavelet coefficient tree is described in more detail in chapter 6. In this section I will outline the major steps of the algorithm. First, the common wavelet coefficient tree is built for all principal components – each principal component is represented by exactly the same tree. Thus, in each tree node required coefficients are stored for all principal components. In fact there are three separate coefficients (red, green, and blue) for each principal component. The advantage of having one common coefficient tree is that in order to evaluate a specific BRDF value only one tree traversal is required. Each BRDF (original or synthesized) can be represented by n coefficients for the principal components (e.g., $n = 45$); thus, during the tree traversal instead of accumulating one real value I need to accumulate vectors in $\mathbb{R}^{3n}$ (red, green, and blue value for each component). As a result, in one tree traversal I obtain the red, green, and blue value for each of the n principal components. The final values of red, green, and blue components of a BRDF are obtained by linearly combining principal component values weighted by the coefficients describing the specific BRDF.

The shader used in Monte Carlo path tracing requires both the BRDF and the importance function. The common wavelet coefficient tree compresses only the reflectance function, and an analogous representation is required to efficiently store the importance functions. Lalonde [26] shows that the importance functions can be also efficiently represented using wavelets. Therefore, in principle, similar analysis could be performed to compress the basis importance functions using wavelets. Alternatively, some prototypical importance functions could be compressed using wavelets. Then, a particular BRDF could have assigned one of these functions – the one that is the most similar to the optimal importance function. However, this particular shader has not been implemented and is left as future work.

A.4 Summary

In this appendix I have described how to use the data-driven reflectance model in order to render realistic images. I have incorporated my reflectance model into a global illumination renderer. I have described three shaders that trade off realism, time of execution, and memory requirements. The first shader works only with direct illumination but it is very simple and fast. The second shader is used in a Monte Carlo path tracing to obtain a global illumination solution. It is the slowest but produces the most realistic images. The third shader renders pictures directly from wavelet-compressed representation – its memory requirements are the smallest.

Bibliography

[1] G. Beylkin, R. Coifman, and V. Rokhlin, *The fast wavelet transform and numerical algorithms.*, Communications on Pure and Applied Mathematics **44** (1991), no. 2, 141–183.

[2] C. M. Bishop, *Neural networks for pattern recognition*, Clarendon Press, 1995.

[3] V. Blanz and T. Vetter, *A morphable model for the synthesis of 3D faces*, Computer Graphics **33** (1999), no. Annual Conference Series, 187–194.

[4] J. Blinn, *Models of light reflection for computer synthesized pictures*, Computer Graphics **11** (1977), no. Annual Conference Series, 192–198.

[5] M. Brand, *Charting a manifold*, Advances in Neural Information Processing Systems, vol. 15, 2003.

[6] W-C. Chen, J-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, *Light field mapping: Efficient representation and hardware rendering of surface light fields*, ACM Transactions on Graphics **21** (2002), no. 3, 447–456.

[7] R. Cook and K. Torrance, *A reflection model for computer graphics*, ACM Transactions On Graphics **1** (1982), no. 1, 7–24.

[8] *Cornell light measurement laboratory*, http://www.graphics.cornell.edu/research/measure/.

[9] O. G. Cula and K.J. Dana, *Compact representation of bidirectional texture functions*, Proceedings of Computer Vision and Pattern Recognition, 2001, pp. 1041–1047.

[10] *Curet: Columbia-utrech reflectance and texture database*, http://www.cs.columbia.edu/CAVE/curet/.

[11] K. Dana, *BRDF/BTF measurement device*, Proceedings of IEEE International Conference on Computer Vision, 2001, pp. 460–466.

[12] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, *Reflectance and texture of real world surfaces*, ACM Transactions on Graphics **18** (1999), no. 1, 1–34.

[13] P. Debevec and J. Malik, *Recovering high dynamic range radiance maps from photographs*, Computer Graphics **31** (1997), no. Annual Conference Series, 369–378.

[14] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, Wiley-Interscience Publication, 1973.

[15] R. W. Floyd, *Algorithm 97: Shortest path.*, Communications of the ACM **5** (1962), no. 6, 345.

[16] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, *The lumigraph*, Computer Graphics **30** (1996), no. Annual Conference Series, 43–54.

[17] X. He, K. Torrance, F. Sillon, and D. Greenberg, *A comprehensive physical model for light reflection*, Computer Graphics **25** (1991), no. Annual Conference Series, 175–186.

[18] H. W. Jensen, *Realistic image synthesis using photon mapping*, AK Peters, 2001.

[19] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, *A practical model for subsurface light transport*, Computer Graphics **35** (2001), no. Annual Conference Series, 511–518.

[20] I. T. Jolliffe, *Principal component analysis*, Springer-Verlag, 1986.

[21] J. T. Kajiya, *Anisotropic reflection models.*, Computer Graphics **19** (1985), no. Annual Conference Series, 15–21.

[22] J. Kautz and M. McCool, *Interactive rendering with arbitrary BRDFs using separable approximations*, Rendering Techniques '99, 1999, pp. 281–292.

[23] J. Koenderink, A. van Doorn, and M. Stavridi, *Bidirectional Reflection Distribution Function expressed in terms of surface scattering modes*, Proceedings of European Conference on Computer Vision, 1996.

[24] J. B. Kruskal and M. Wish, *Multidimensional scaling*, Sage Publications, 1978.

[25] E. Lafortune, S.-C. Foo, K. Torrance, and D. Greenberg, *Non-linear approximation of reflectance functions*, Computer Graphics **31** (1997), no. Annual Conference Series, 117–126.

[26] P. Lalonde, *Representations and uses of light distribution functions*, Ph.D. thesis, The University of British Columbia, 1997.

[27] P. Lalonde and A. Fournier, *A wavelet representation of reflectance functions*, IEEE Transactions on Visualization and Computer Graphics **3** (1997), no. 4, 329–336.

[28] D. Lee and H. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature **401** (1999), 788–791.

[29] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, *Image-based reconstruction of spatially varying materials*, Rendering Techniques '01, 2001, pp. 104–115.

[30] R. Lu, J. Koenderink, and A. Kappers, *Optical properties (Bidirectional Reflectance Distribution Functions) of velvet*, Applied Optics **37** (1998), no. 25, 5974–5984.

[31] D.W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, Journal of the Society for Industrial and Applied Mathematics **11** (1963), 431–441.

[32] S. Marschner, *Inverse rendering for computer graphics*, Ph.D. thesis, Cornell University, Ithaca, NY, 1998.

[33] S. Marschner, S. Westin, E. Lafortune, and K. Torrance, *Image-based measurement of the Bidirectional Reflection Distribution Function*, Applied Optics **39** (2000), no. 16, 2592–2600.

[34] S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg, *Image-based brdf measurement including human skin*, Rendering Techniques '99, 1999, pp. 139–152.

[35] D. K. McAllister, A. Lastra, and W. Heidrich, *Efficient rendering of spatial bidirectional reflectance distribution functions*, Graphics Hardware, 2002, pp. 79–88.

[36] M. McCool, J. Ang, and A. Ahmad, *Homomorphic factorization of BRDFs for high-performance rendering*, Computer Graphics **35** (2001), no. Annual Conference Series, 171–178.

[37] F. Nicodemus, J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis, *Geometric considerations and nomenclature for reflectance*, Monograph 160, National Bureau of Standards (US), October 1977.

[38] *Nist reference reflectometer: Starr facility*, http://physics.nist.gov/.

[39] M. Oren and S. K. Nayar, *Generalization of lambert's reflectance model*, Computer Graphics **28** (1994), no. Annual Conference Series, 117–126.

[40] F. Pellacini, J. Ferwerda, and D. Greenberg, *Toward a psychophysically-based light reflection model for image synthesis*, Computer Graphics **34** (2000), no. Annual Conference Series, 55–64.

[41] B. T. Phong, *Illumination for computer generated pictures*, Communications of ACM **18** (1975), no. 6, 311–317.

[42] S. Roweis and L. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science **290** (2000), no. 5500, 2323–2326.

[43] S. Rusinkiewicz, *A new change of variables for efficient BRDF representation*, Rendering Techniques '98, 1998, pp. 11–22.

[44] Y. Sato, M. D. Wheeler, and K. Ikeuchi, *Object shape and reflectance modeling from observation*, Computer Graphics **31** (1997), no. Annual Conference Series, 379–387.

[45] P. Schröder and W. Sweldens, *Spherical wavelets: Efficiently representing functions on the sphere*, Computer Graphics **29** (1995), no. Annual Conference Series, 161–172.

[46] E. Stollnitz, T.Derose, and D.Salesin, *Wavelets for computer graphics*, Morgan Kaufmann Publishers, 1996.

[47] G. Strang and T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.

[48] Y. W. Teh and S. T. Roweis, *Automatic alignment of hidden representations*, Advances in Neural Information Processing Systems, vol. 15, 2003.

[49] J. Tenenbaum, V. de Silva, and J. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science **290** (2000), no. 5500, 2319–2323.

[50] K. Torrance and E. Sparrow, *Theory for off-specular reflection from roughened surfaces*, Journal of the Optical Society of America **57** (1967), no. 9, 1105–1114.

[51] K. Torrance, E. Sparrow, and R. Birkebak, *Polarization, directional distribution, and off-specular peak phenomena in light reflected from roughened surfaces*, Journal of the Optical Society of America **56** (1966), no. 7, 916–925.

[52] T. Trowbridge and K. Reitz, *Average irregularity representation of roughened surfaces*, Journal of the Optical Society of America **65** (1975), no. 5, 531–536.

[53] V. Vapnik, *The nature of statistical learning theory*, Springer, 1995.

[54] E. Veach, *Robust monte carlo methods for light transport simulation*, Ph.D. thesis, Stanford University, 1997.

[55] G. Ward, *Measuring and modeling anisotropic reflection*, Computer Graphics **26** (1992), no. Annual Conference Series, 265–273.

[56] S. Westin, J. Arvo, and K. Torrance, *Predicting reflectance functions from complex surfaces*, Computer Graphics **26** (1992), no. Annual Conference Series, 255–264.

[57] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, *Inverse global illumination: Recovering reflectance models of real scenes from photographs*, Computer Graphics **33** (1999), no. Annual Conference Series, 215–224.

[58] Z. Zhang, *A flexible new technique for camera calibration*, Tr98-71, MSR Redmond, 1998.