Fast Extraction of BRDFs and Material Maps from Images

by

Rafal Jaroszkiewicz

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Mathematics

in

Computer Science

Waterloo, Ontario, Canada, 2003

Author's Declaration for Electronic Submission of a Thesis

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The bidirectional reflectance distribution function has a four dimensional parameter space and such high dimensionality makes it impractical to use it directly in hardware rendering. When a BRDF has no analytical representation, common solutions to overcome this problem include expressing it as a sum of basis functions or factorizing it into several functions of smaller dimensions.

This thesis describes factorization extensions that significantly improve factor computation speed and eliminate drawbacks of previous techniques that overemphasize low sample values. The improved algorithm is used to calculate factorizations and material maps from colored images. The technique presented in this thesis allows interactive definition of arbitrary materials, and although this method is based on physical parameters, it can be also used for achieving a variety of non-photorealistic effects.

Acknowledgements

I would like to thank my supervisor, Professor Michael McCool, for his support, guidance, and encouragement. I am very grateful to him for sharing his knowledge and for teaching me the art and science of computer graphics.

I also would like to thank my readers, Professor Stephen Mann and Professor Peter Forsyth, for taking time from their busy schedule to review my thesis and provide invaluable comments.

Special thanks to Selina Siu for discussions, suggestions, and for painting sample images. Models used in this thesis have been provided by Alla Sheffer and Kevin Moule.

Thanks to the professors and students of the Computer Graphics Lab, for providing an incredibly warm and friendly atmosphere, and for making these years unforgettable fun.

This research was funded by grants from the National Science and Engineering Research Council of Canada (NSERC), the Centre for Information Technology of Ontario (CITO), the Canadian Foundation for Innovation (CFI), the Ontario Innovation Trust (OIT), the Bell University Labs initiative, and ATI Technologies.

Dedication

To my family, to Ana, and to my friends.

Contents

1 Introduction **1**

2 Background **5**

2.1 BRDF and Radiometric Terms 6

2.2 BRDF Representation 10

2.3 BRDF Acquisition 11

2.4 Texture Shading 13

2.5 Previous Work 14

3 Extending Homomorphic Factorization **16**

3.1 Homomorphic Factorization 17

3.2 Extensions 21

3.2.1 1D Factors 21

3.2.2 Preinverted Matrix 23

3.2.3 Mapping Between Linear and Log Spaces 26

3.3 Factorization on Graphics Hardware 29

3.4 Summary 30

4 Recovering BRDFs from Images **31**

4.1 Spatially Uniform BRDFs 33

4.2 Spatially Varying BRDFs 35

4.2.1 Four Factor Product 35

4.2.2 Material Mapping Recovery 37

4.3 Hybrid Method 42

4.4 Integration of Multiple Views 42

4.5 Summary 43

5 Results **45**

5.1 Discussion 50

6 Conclusion **58**

6.1 Future Work 59

Bibliography **62**

List of Tables

5.1 SVD computation times in seconds for preinversion and factorization (1D factors) using 7884 samples. 46
5.2 SVD computation times in seconds for preinversion and factorization (2D factors) using 317 samples. 47
5.3 Material map computation time in seconds. 47
5.4 Error metrics for 4-factor approximation and material map approximations using between 1 and 5 product components. 47

List of Figures

2.1 Incoming and outgoing light rays in a local frame 7

2.2 Isotropic and anisotropic materials 9

2.3 Difference between uniform and non uniform materials 9

3.1 Directional vector projections 21

3.2 Factors for an isotropic material 22

4.1 Interpolation of the coefficient samples 40

5.1 Phong reflectance model and its factorization 54

5.2 Painted and rendered duck model. 54

5.3 The effect of smoothing coefficient on the approximation 54

5.4 A painted sphere and a cat under changing light and view direction. 55

5.5 Capturing the spatially varying materials 55

5.6 SBRDF recovery using multiple views 56

5.7 A photograph of a sphere wrapped in a velvet material. 56

5.8 Examples of painted and recovered shading models. 57

Chapter 1

Introduction

In this thesis we investigate two issues:

- improving the representation and computation of BRDFs in factorized forms
- extracting reflectance models from digitally painted images.

Although these are separate problems, the technique of obtaining shading models (the second issue), as presented in this thesis, is based on the concepts of extended homomorphic factorization (the first issue).

Factorization is primarily used for high performance hardware rendering of complex reflectance models using textures as lookup tables of sampled functions. In this approach, a high dimensional BRDF is approximated by a product of several functions of smaller dimensions stored in textures that are used for shading during a rendering stage. Factorized representation works well in practice because many empirical and theoretical reflectance models (e.g., Phong, Cook-Torrence, Ashikhmin) are analytically factorizable. Homomorphic factorization is a tech-

nique for obtaining the factor functions from BRDF samples. The factors give the best approximation to the BRDF in the least square residual sense.

The first part of this thesis presents extensions to homomorphic factorization [20]. One of the extensions permits fast computation of approximations to BRDFs by precomputing a pseudo-inverse of the constraint matrix, which is a part of a homomorphic factorization detailed in chapter 3. It is possible to reduce factorization time to less than a few seconds (a speedup of a factor of over 1000). The precomputation time is still on the order of several minutes, but consecutive calculations of new factors can be performed at interactive rates. This improved computation time is crucial for interactive applications and is exploited in our painting system described later.

Another extension allows a more compact representation of factors for a special case of isotropic materials. For this type of materials, the factors can be represented by 1D instead of 2D functions. This improvement conserves scarce hardware resources on graphics cards. It also enables a denser sampling of factor functions yielding better precision. The smaller representation also reduces the computation time even further.

While the original homomorphic factorization tends to emphasize small BRDF sample values, alternative mappings from linear to logarithmic space give better control over this selectiveness. One of the mappings, presented in this thesis, assigns roughly equal significance to both low and high valued samples. Thus, the factorization does not ignore important information and produces more accurate approximations. This is especially important for materials that have a dark base

color and a bright, sharp specular peak. In those situations, a majority of samples will represent small reflectance values of the diffuse part, and only a few samples will probe the bright highlight. If small samples dominate the computations, the resulting approximation will lack the specular highlight.

The second part of this thesis treats the issue of recovering reflectance models from images. This subject is related to inverse rendering, which deals, in part, with obtaining BRDFs from pictures. Reflectance models acquired this way can be used for rendering faithful replicas of materials captured in original images. Inverse rendering is especially valuable for finding reflectance models of real world materials from photographs. However, one drawback of using photographs to find reflectance models is an inherent lack of flexibility: it is difficult to define a material without having its counterpart in real world.

An easy solution to this problem is to use digital painting to modify existing photographs or create new images from scratch. This allows the freedom to define an arbitrary reflectance model by painting an image. This thesis presents a technique that integrates the extended homomorphic factorization with a painting application to create a system for interactive definition of reflectance models. In such an application, the user is presented with the scene setup (a view of a 3D object and a light position) and is allowed to color the image to define the appearance of the object. Our new factorization algorithm is then invoked to calculate the BRDF approximation from the painted images. The system is capable of recovering a variety of reflectance models ranging from non-photorealistic to physically based ones, including multi-material composition and spatially varying BRDFs. The non-

homogeneous surfaces are approximated using material maps that combine several BRDFs using texture mapping as blending coefficients. The final outcome of the process is a set of BRDF and material maps that can be retargeted to any other model under any view or lighting directions and rendered in real time on current hardware.

In Chapter 2 we present background information and previous work related to reflectance models and measuring BRDFs. Chapter 3 summarizes homomorphic factorization and describes extensions that allow fast calculations. The method of using painted images for BRDF definition is presented in Chapter 4. We show and discuss the results in Chapter 5 and conclude the paper in Chapter 6.

Chapter 2

Background

The appearance of an object is determined by the properties of the material of which it made of. The material properties essentially define how light interacts with the object when light falls on a surface. Usually a portion of the light is absorbed while the rest is scattered in various directions to be perceived by an observer as colors. In computer graphics the surface properties are modeled with a bi-directional reflectance distribution function (BRDF). In essence, it defines how much light arriving from an incoming direction is reflected in a given outgoing direction.

Even though the BRDF models closely the appearance of an object, it is still only an approximation to the behavior of light. For example, it assumes that the light leaves a surface from the same point that it arrives. In real materials, however, there is some degree of subsurface scattering before the light exits the object at a point other than it entered. There are also other factors that influence the reflection, like polarization, frequency shift, and time delay. Furthermore the BRDF is dependent

on wavelength, which is assumed implicitly. Despite its incompleteness, the BRDF provides an accurate description of surfaces, and we will accept this approximation as one of our assumptions.

2.1 BRDF and Radiometric Terms

Before a formal definition of the BRDF is presented, it is helpful to review the radiometric quantities that describe light.

Light is a flow of energy in the form of electromagnetic radiation. The amount of energy flowing through a cross sectional plane in a given amount of time is called the *radiant flux*, $\Phi = \frac{dW}{dt}$. Although power is a quantity without direction, radiant flux has a direction that is used to determine other measurements of light.

Irradiance is used to characterize the density of the flux passing though (or falling on) a given area. It is defined as $E = \frac{d\Phi}{dA}$ where $d\Phi$ is the radiant flux and dA is the area.

Irradiance measures the density of light falling on a surface from all directions (unless explicitly specified otherwise). In turn, *radiance* measures the irradiance per solid angle for a given direction: $L = \frac{d^2\Phi}{d\omega\, dA_\perp}$, where $d\omega$ is a differential solid angle and $dA_\perp$ is a projected area of a surface in the direction of the light ray. *Radiance* is important, because it translates into a number attached to a ray in a ray-tracer.

An object can be perceived as a blob of colors only if it emits or reflects light. Reflective color is determined by the surface characteristics that are approximated mathematically in computer graphics by the bi-directional reflectance distribution

function, commonly known as the BRDF. It describes the proportion of light arriving from one direction $\hat{\omega}_i$ that is scattered in another direction $\hat{\omega}_o$:

$$f_r(\hat{\omega}_i, \hat{\omega}_o) = \frac{dL_o(\hat{\omega}_o)}{dE(\hat{\omega}_i)} = \frac{dL_o(\hat{\omega}_o)}{L_i(\hat{\omega}_i)\, cos\theta_i\, d\omega_i} \tag{2.1}$$

Figure 2.1 illustrates the concept of a BRDF. It shows incoming light direction $\hat{\omega}_i = (\phi_i, \theta_i)$, the outgoing viewing direction $\hat{\omega}_o = (\phi_o, \theta_o)$, with respect to the local frame $\mathcal{F} = (\mathbf{x}, \mathbf{y}, \mathbf{z})$. The differential solid angle encompassing incoming light is denoted by $d\omega_i$.

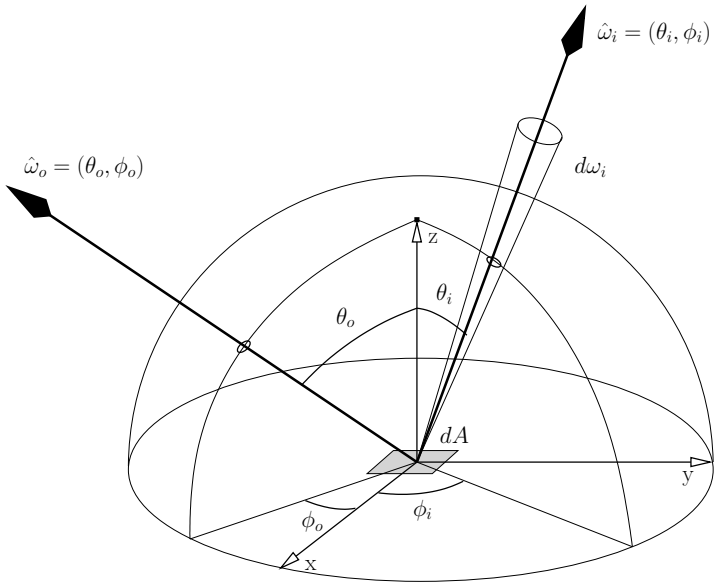


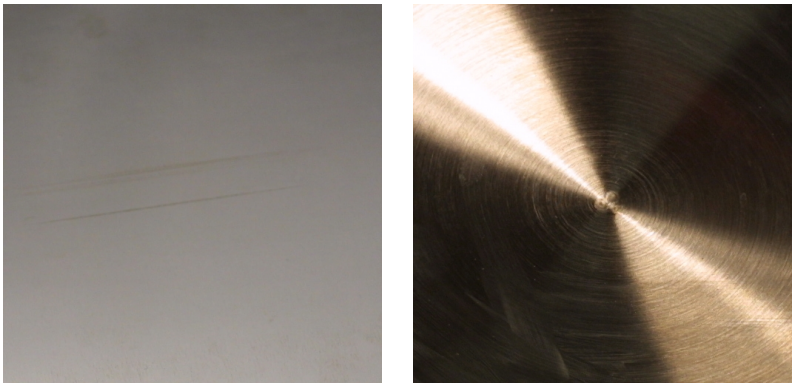
Figure 2.1: Incoming and outgoing light rays in a local frame.

In real world, the BRDF has three well known properties. It is reciprocal, which means that if we switch the light and view direction arguments then the BRDF value stays the same (for non-singular functions). Another property is that the BRDF is

always positive. The third characteristic follows from energy conservation principle and guarantees that the integral over the outgoing hemisphere is always less than one, although locally a BRDF is not bounded from above. This indicates that amount of reflected light energy cannot be greater than the amount of incoming light. In computer graphics BRDFs do not always satisfy these three conditions; if they do, they are said to be physically plausible.

One important class of BRDFs, called isotropic, is invariant with regard to the rotation of the surface around its normal with the view and light directions fixed with respect to global reference frame. Isotropic reflectance models are common in the real world, and because they do not require local coordinate frames on the surfaces (except the normal), they are popular in computer graphics. This type of BRDF does not depend on ϕ_i and ϕ_o explicitly, but rather on $\phi_i - \phi_o$ which makes them a 3D function. Other BRDFs that explicitly depend on the values of ϕ_i and ϕ_o are called anisotropic. This type of material occurs because of the existence of preferential direction in the microgeometry structure of the surface. The difference in the visual appearance of isotropic and anisotropic disks is shown in Figure 2.2.

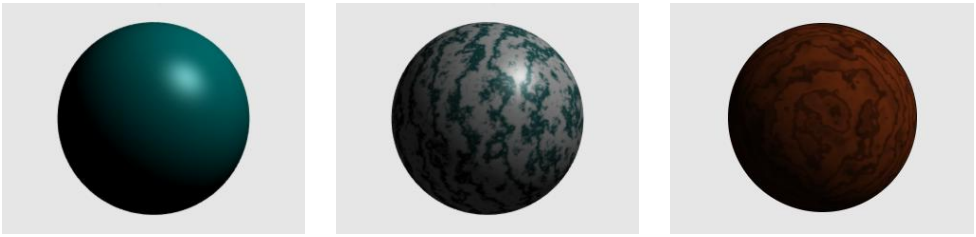
When an object is made of a uniform material, one BRDF approximates the light behavior at every point on its surface. Hoverer, most of the objects in real life exhibit reflectance variations over their surfaces. The reason is that different areas of one object are made of different materials with different optical properties, or are made of one with different "finishes". Figure 2.3 shows spheres made of plastic, a uniform material that can be modeled with one BRDF, and marble, an inhomogeneous material that requires two BRDFs. The third image shows



(a) (b)

Figure 2.2: Isotropic and anisotropic materials: (a) isotropic smooth metal, (b) anisotropic brushed metal

wood whose BRDFs have smooth transitions. To model the general surface-light interaction of nonuniform objects, it is necessary to expand the concept of the BRDF to include the spatial dependence of the reflectance. Such an extended function, sometimes called a spatial BRDF (SBRDF) or a bi-directional texture function (BTF), has six parameters: two for incoming direction of light, two for outgoing direction of light, and two to locate a point on the object's surface.



(a) (b) (c)

Figure 2.3: Difference between uniform and non uniform materials: uniform material (a), and varying materials - marble (b) and wood (c)

2.2 BRDF Representation

A straightforward way to represent an arbitrary BRDF is to tabulate its sampled values. To calculate the function value for any arguments, the entries in a table are interpolated using linear interpolation or another scheme. Despite the advantage of flexibility and simplicity, this approach of storing BRDFs is not popularly used. One of the major drawbacks is the storage size requirements associated with storing a 4D function. To accurately represent the BRDF, it must be densely sampled, which combined with the high dimensionality inflates the storage to forbidding size. Another disadvantage is the missing data and necessity of interpolating over large parameter distances. Due to noisy samples, large residual errors of some measurements affect the final function value.

One way to overcome those problems is to fit the BRDF to a theoretical model expressed by a mathematical formula. Usually mathematical models yield a good approximation to popular types of BRDFs. Because models have a small number of free parameters, fitting samples to them gives stable and well behaved results. Also this representation is compact since all that is stored are the free parameters. A disadvantage of this approach is the lack of flexibility, because this representation cannot capture all BRDFs, due to a limited number of parameters. Also some of the theoretical models are complicated and expensive to evaluate. However, if the formulation is simple and has a sufficient number of parameters, this can be a powerful representation. There are many popular reflectance models that belong to this representation category; among them a well known Lafortune model that is used widely in computer graphics. A sum of lobes from a suitably flexible model,

such as the Lafortune model, can also be considered a general representation. In this case, the constants describing each of the lobes are regarded as reflectance model parameters. Unfortunately, calculating these parameters is usually hard and involves solving a large, non-linear, error minimization, fitting problem.

A third method to encode BRDFs is to use basis functions like spherical harmonics, wavelets, or Zernike polynomials (on a disk). Finding the correct representation for a given BRDF consists of finding the appropriate coefficients. Orthogonal basis functions, such as the spherical harmonics, are especially useful, since finding the coefficients requires only a projection of the data against the basis functions. There are several advantages of basis function representation including compactness and ease of computation. However, complex BRDFs may need many basis functions to be adequately approximated, and for example, higher order spherical harmonics become increasingly expensive to evaluate. Also basis functions are not always spatially localized which results in ringing when they are fitted to data. This ringing is known as the Gibbs phenomenon (Figure 5.3).

2.3 BRDF Acquisition

One method of BRDF measurement is the theoretical model simulation. In this method, a surface microgeometry model is devised and later used during a ray tracing simulation. Ray tracing usually takes into considerations all physical properties of light and materials, such as, reflection, subsurface scattering, and polarization. During simulation, rays are shot from an incoming direction, and for each exitant direction the number of reflected rays is counted. The density and power of the

reflected rays are used to estimate the BRDF for a given surface model [1, 24].

Such an approach to measuring BRDFs is easy to implement and gives consistent results. It requires, however, an accurate physical surface model that is usually hard to formulate, especially if wave effects are to be included. Also, usually having built the surface model it is straight forward to derive a reflection model which will represent the BRDF. But, again, the simulation can precompute the values if the mathematical formula for the BRDF is expensive to evaluate.

A more empirical approach of finding a BRDF is to take physical measurements for densely sampled directions of incoming and outgoing light. Although this approach yields realistic BRDFs, it is usually error prone and so the results include large measurement errors. The difficulty of conducting accurate measurements comes from many factors such as instability of the apparatus assembly and specimen inhomogeneity. Nevertheless, these measurements are important for successful synthesis of photorealistic images due to their connection with real world [2, 17, 19].

The last method of defining reflection models is based on an artist's skills where an interactive painting tool is used to define the object's appearance. This approach is widely used in non-photorealistic rendering, and is currently popular for defining an arbitrary texture. Paint strokes usually change the color sample (i.e., the pixel value), but a virtual brush may also be used to indirectly modify the parameters of a lighting model. Usually painting is used to define functions that are only a subset of the full BRDF function, for example, using the Lambertian reflectance model (that depends only on light vector), or BRDFs approximated by some analytical

reflectance model. Also, painting may not define a physically plausible BRDF, due to a possible lack of reciprocity and energy conservation. Nevertheless, this method is flexible and is suitable for subjective rendering techniques [5, 22].

2.4 Texture Shading

When rendering objects, the BRDF is used to compute color based on relative light position and view direction. In interactive applications, the computations must be fast to maintain the real-time frame rates for smooth animation. One way of increasing the performance is to precompute certain values and exploit texture hardware on current graphics cards for storage. Texture is thought of as a 2D array optimized for quick access and value interpolation. Before the repeated rendering of a scene, it is often possible to precompute colors or other values and encode them in a texture as a sampled 2D function. During rendering the texture lookups of precomputed values replace lengthy computations. It may be still necessary to perform a small amount of calculations to combine the results of several lookups, but nevertheless, the computational cost is significantly reduced.

A major advantage of texture shading is its speed and flexibility. Also filtering (mip-mapping) can be used to avoid aliasing. The disadvantages are large storage requirements and unavoidable errors in sampling, quantization, and interpolation.

2.5 Previous Work

The idea of representing BRDFs by several functions of smaller dimensions was introduced by Fournier [3], who used numerical factorization, and Heidrich and Seidel [6], who used analytic factorization. Kautz and McCool [10] used separable functions and the SVD to approximate reflectance models. McCool, Ang and Ahmad [20] presented a factorization method that solved the drawbacks of the previous methods. McCool, Ang and Ahmad's approach, homomorphic factorization, permits the factorization of functions of arbitrary dimension into products of several functions of smaller dimensions and the use of arbitrary parameterizations.

Other methods of BRDF representation rely on the summation of basis functions, for example, spherical harmonics [21]. Malzbender, Gelb and Wolters [16] used a polynomial basis to reconstruct surface color under varying light direction. This method captures spatial variation of color on surfaces [11, 14] and is equivalent to the use of spherical harmonics. McAllister, Lastra and Heidrich [19] fitted Lafortune lobes [12] to measured data and stored the coefficients in texture maps, allowing real-time rendering of spatially varying BRDFs.

Lensch et al. [14] were successful in recovering spatially variant materials by grouping samples into clusters representing a single material. More recently Hertzmann and Seitz [7] developed their algorithm of material recovery based on clustering method. An advantage of these algorithms is the high accuracy of the captured materials they produce; however their disadvantage is a long computational time.

Measured data was used by Marschner et al. [17] to find reflectance models of human skin [2]. Our work borrows from their approach, but differs in BRDF repre-

sentation, and, instead of photographing the objects, we use software for painting models. Painting has been explored by Hanrahan and Haeberli [5] and Kalnins et al. [9] to interactively paint onto a parametrized 3D model. Sloan et al. [22] also explored painting as a method to capture shading for non-photorealistic rendering. They use projection of the surface normal onto the viewing plane as a shading parameter; in contrast, our method incorporates the view vector, the light vector and texture coordinates as function parameters, thus allowing greater flexibility. In fact, our method can be extended to any BRDF parameterization like the original homomorphic factorization method.

Chapter 3

Extending Homomorphic Factorization

As mentioned earlier, one of the BRDF representation methods is *factorization*. This technique approximates a function with a product of several low dimensional functions that can be either analytically derived or numerically calculated. One of the numerical algorithms is homomorphic factorization which computes suitable factors used to approximate 4D reflectance (or other) functions. In this chapter we will describe homomorphic factorization and give details on our extensions to this method.

3.1 Homomorphic Factorization

Homomorphic factorization is based on an assumption that a function of interest can be closely approximated by a product of other functions:

$$f(\mathbf{x}) \approx \prod_{j=1}^{J} p_j(\pi_j(\mathbf{x})), \tag{3.1}$$

where π_j are mappings from a domain of the function f to the domains of functions p_j. By taking a logarithm $\tilde{a} = \log(a)$ on both sides of the above approximation, we obtain a linear equation:

$$\tilde{f}(\mathbf{x}) \approx \sum_{j=1}^{J} \tilde{p}_j(\pi_j(\mathbf{x})) \tag{3.2}$$

where $\tilde{f} = \log(f)$ and $\tilde{p}_j = \log(p_j)$. The logarithm will compress the range of data for large values of f and expand it for small values of f. Issues related to this nonlinear mapping will be discussed in Section 3.2.3.

The factors p_j are functions whose one- or two-dimensional domains are discretized and whose values are sampled into lookup tables. Our goal, then, is to find appropriate entries (i.e., sampled function values) in those tables. If we have N samples of function $\tilde{f}$, we can set up a system of N linear equations with the entries in the lookup tables as unknowns. To describe this system of equations in a matrix form, we need to convert 2D M by M factor tables into a 1D vector $\mathbf{y}$. This is easily done by scanning the table entries row by row an assigning them to the consecutive elements of the vector. The samples $\tilde{f}$ are related to the unknown vector $\mathbf{y}$ by Equation 3.2. The matrix representing this equation can be

decomposed into J submatrices, each of which applies to one factor. We will use the projection mappings π_j to construct the submatrices A_j that select which entry of the table (vector) the direction $\mathbf{x}$ is projected for a given sample. For greater accuracy, A_j will actually select four closest table (vector) entries and bilinearly interpolate them to obtain a better approximation. All but four entries in each row of a N by M^2 matrix A_j are zero, and so the matrix A_j is sparse. We can describe the Equation 3.2 using matrix notation:

$$\left[\tilde{\mathbf{f}} \right] = \left[\mathsf{A}_1 \quad \cdots \quad \mathsf{A}_J \right] \begin{bmatrix} \tilde{\mathbf{p}}_1 \\ \vdots \\ \tilde{\mathbf{p}}_J \end{bmatrix}. \tag{3.3}$$

A typical problem with samples is noise. Also, there may not be enough samples available to relate each element of the vector of unknowns to the sample values, leading to an underdetermined system of equations. To avoid the noise problem, we can introduce smoothness constraints on the factor functions. If a function is smooth with low curvature, its convolution with a Laplacian filter will be close to zero, because Laplacian is a second order derivative calculation operator. We can incorporate the smoothness condition into Equation 3.3 by equating to zero the Laplacian convolution with the factor function for each table entry. Thus, for each vectorized table (i.e., matrix element $\tilde{\mathbf{p}}_j$) we will construct a M^2 by M^2 matrix L_j, whose rows describe a convolution for one lookup table entry. By equating these convolutions to zero, we impose low curvature on the factor functions, reducing ripples and noise. We can also use scaling factor λ for each matrix L_j to adjust the

importance of smoothness, with large values of λ forcing the solution to be smooth. Incidently, the smoothness constraint also solves the problem of insufficient number of samples because now all the table entries are constrained in some way, and the matrix is guaranteed to be of full rank.

In a summary, to perform homomorphic factorization [20] of a BRDF (or other high-dimensionality function) it is necessary to solve, in a minimum-residual or least-squares sense, an overconstrained linear system of the form

$$\begin{bmatrix} \tilde{\mathbf{f}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathsf{A}_1 & \mathsf{A}_2 & \cdots & \mathsf{A}_J \\ \lambda\mathsf{L}_1 & & & \\ & \lambda\mathsf{L}_2 & & \\ & & \ddots & \\ & & & \lambda\mathsf{L}_J \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{p}}_1 \\ \tilde{\mathbf{p}}_2 \\ \vdots \\ \tilde{\mathbf{p}}_J \end{bmatrix}, \tag{3.4}$$

where $\tilde{\mathbf{f}}$ is a vector of data points to be fitted (logarithms of sample values of an N-dimensional function f), the $\tilde{\mathbf{p}}_j$ are logarithms of samples of lower-dimensional functions p_j (which f project onto), the matrices A_j describe the projections of samples onto each factor as a set of constraints, and finally the matrices L_j are Laplacian operators that estimate the curvature of the functions p_j. This overdetermined system of linear constraints can be written more compactly as $\mathbf{g} = \mathsf{B}\mathbf{y}$.

The theoretical description of homomorphic factorization presented so far has an immediate practical application in computer graphics which was also described in the original paper [20]. Three factors p, q, and r were used to approximate a BRDF from samples. The choice of three factors and the associated projection functions

π_p, π_q, and π_r was based on popular reflectance models that take into account the microfacet distribution, and the Fresnel effect, as well as masking and shadowing effects. The microfacet distribution depends on a half vector between light and view direction in a local coordinate frame. The Fresnel coefficient depends on the light direction, and the masking-shadowing effect can be modeled as a geometry term parametrized by the viewing direction. For these reasons the parameter of function p depends on the light vector, q depends on the half vector, and r depends on the view vector. To map a vector direction from a hemisphere to a square domain, a hemispherical, a parabolic, or a cube map can be used. The hemispherical map (see Figure 3.1 a) takes the intersection point of a unit hemisphere and the line defined by the direction vector, and projects it on to the plane orthogonal to the z vector. Parabolic projection (Figure 3.1 b) works in a similar way, but instead of hemisphere, a paraboloid is used. The parabolic map has the advantage of avoiding space compression near the horizon and for that reason it was chosen by McCool, Ang and Ahmad, and I use it in this thesis. An even more uniform mapping can be obtained by assembling a 2D domain into a cube, centering it about the origin, and defining the mapping for a directional vector as a point of the intersection of the cube with the line formed by this vector. Such correspondence is called a cube map (Figure 3.1 c). Another advantage of the cube map is explicit hardware support for it on current graphics cards and better interpolation behavior.

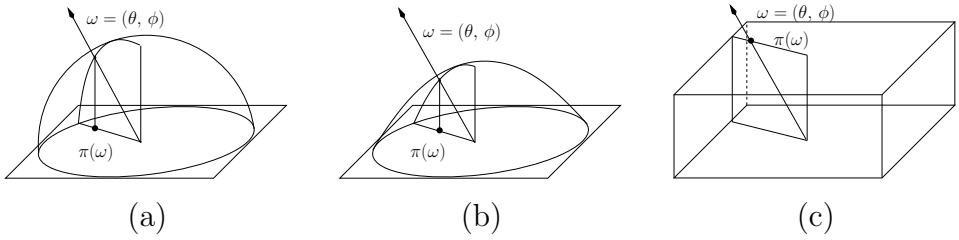


Figure 3.1: Directional vector projections: (a) hemispherical (b) parabolic (c) cube map

3.2 Extensions

The homomorphic factorization can be extended to improve performance and compactness in certain situations. We now describe three extensions that constitute a part of this thesis' contribution.

3.2.1 1D Factors

The application of homomorphic factorization to BRDFs presented by McCool, Ang and Ahmad [20] can handle both isotropic and anisotropic materials. They use a parabolic map to project the BRDF's parameters from four dimensions onto a two dimensional domain of factor functions. However, if a BRDF is isotropic, the factor functions can be proven to be radially symmetric under their parameterization (see Figure 3.2) and thus are really one dimensional. In practice the size of the constraint matrix is a linear function of the number of texels, therefore changing the dimensionality of factors from 2D to 1D yields a large reduction in matrix size. If we know that the BRDF is isotropic, the pseudo-inverse computation time, which depends on the matrix size, can be significantly diminished.

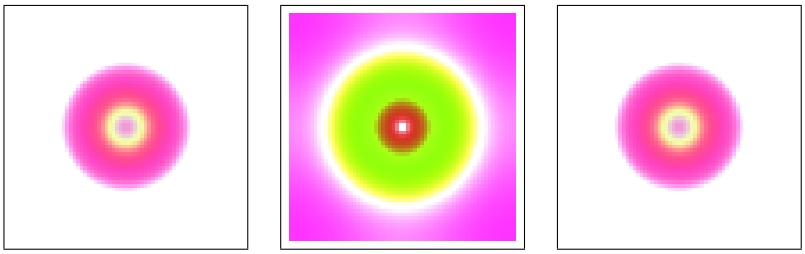


Figure 3.2: Factors yield by homomorphic factorization of an isotropic material.

As described in Section 3.1, given a set of samples, a constraint matrix is formed. To take advantage of isotropic materials, the factors this matrix constrains are going to be one dimensional. As a demonstration, we use three-factor product approximation p, q, and r, where each factor is a 1D function. The projection transformations that map 4D BRDF parameters into a 1D parameter space for each factor are

$$\begin{aligned} \pi_p(\hat{\omega}_i, \hat{\omega}_o) &= |\pi_{par}(\hat{\omega}_i)| \\ \pi_q(\hat{\omega}_i, \hat{\omega}_o) &= |\pi_{par}(\mathbf{h})| \\ \pi_r(\hat{\omega}_i, \hat{\omega}_o) &= |\pi_{par}(\hat{\omega}_o)| \\ \text{where } \pi_{par}(\mathbf{v}) &= (\mathbf{v}\cdot\mathbf{x},\ \mathbf{v}\cdot\mathbf{y}) \,/\, (1+\mathbf{v}\cdot\mathbf{z})\,, \end{aligned} \tag{3.5}$$

where $\mathbf{h} = \frac{\hat{\omega}_i+\hat{\omega}_o}{|\hat{\omega}_i+\hat{\omega}_o|}$, and $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ are local frame vectors. These transformations return the length of the directional vector projected on the plane orthogonal to the surface normal $\mathbf{n}$ using a parabolic projection. The tail of the vector is at the same point as the center of concentric rings in Figure 3.2, and so the vectors of the same length after projection will correspond to the same factor value.

3.2.2 Preinverted Matrix

Another extension to homomorphic factorization deals with the matrix B that defines the constraining system of equations. For a 3-factor product approximation using 2D factors sampled with a resolution of 32×32 texels, the dimensions of the constraint matrix B are $(number\,of\,samples + 3 \times 32^2)$ by (3×32^2) where $number\,of\,samples$ is typically around $10,000$. The minimum-residual solution of this sparse constraint system can be obtained by iterative methods. However, this is slow and only gives one solution for one set of data. The solution of the system (in the least-squares sense) can also be found using the pseudo-inverse B^+ of B. The pseudo-inverse can be computed using a matrix factorization, for instance the singular value decomposition (SVD):

$$\mathsf{B} = \mathsf{U}\mathsf{S}\mathsf{V}^T = \sum_{i=0}^{N-1} s_i \mathbf{u}_i \mathbf{v}_i^T \tag{3.6}$$

$$\mathsf{B}^+ = \mathsf{V}\mathsf{S}^{-1}\mathsf{U}^T = \sum_{i=0}^{N-1} \frac{1}{s_i} \mathbf{v}_i \mathbf{u}_i^T \tag{3.7}$$

Since the singular values can go to zero for singular matrices, this summation is usually truncated for sufficiently small values of s_i. Small singular values can also be damped out for greater stability [15],

$$\mathsf{B}^+ \approx \sum_{i=0}^{N-1} \frac{s_i}{s_i^2 + \alpha^2} \mathbf{v}_i \mathbf{u}_i^T, \tag{3.8}$$

but this does not give as much computational advantage as simple truncation. A compromise would be to linearly damp out higher singular values. Let $\mathrm{pos}\,(x) =$

$\max(x, 0)$. Then compute the linearly damped SVD with

$$\mathsf{B}^{+} \approx \sum_{i=0}^{N-1} \frac{\mathrm{pos}\,(1 - i\,\gamma)}{s_i} \mathbf{v}_i \mathbf{u}_i^T, \tag{3.9}$$

where γ is an arbitrary coefficient.

Damping singular values has the effect of further smoothing the solution, in addition to the Laplacian constraints. Excessive smoothing can override data, and, thus, special care must be taken not to cause the loss of important information. The smoothing can be controlled by choosing appropriate values for λ and γ, which are usually determined empirically. The effect of smoothing is demonstrated in Chapter 5.

In practice most of the singular values do not diminish to zero, and due to the unique smoothing constraints for each table entry, $col(\mathsf{B})$ usually has a full rank. Thus, we may consider a sparse QR factorization [18] of the matrix B instead of the SVD. The QR factorization may be faster than the SVD, if the algorithm takes advantage of the sparsity. The sparsity may also allow for more compact representation of the factorized matrix.

Factorization As Change of Basis

Finding the BRDF approximation from the sample vector can be viewed as a change of basis, much like the application of spherical harmonics. Consider the computation of the approximate solution $\mathbf{y}^*$ in detail:

$$\mathbf{y}^* = \mathsf{B}^{+}\mathbf{g} = \sum_{i=0}^{N-1} \frac{\mathrm{pos}\,(1 - i\,\gamma)}{s_i} \mathbf{v}_i \mathbf{u}_i^T \mathbf{g}. \tag{3.10}$$

Note that $\mathbf{u}_i^T\mathbf{g} = \mathbf{u}_i \cdot \mathbf{g}$, and the result is a scalar. However, in our particular problem, many entries of $\mathbf{g}$ are zero (due to the Laplacian constraints), so we can just omit that part of $\mathbf{u}_i$ from the summation involved in the computation of $(\mathbf{u}_i \cdot \mathbf{g})$. Let $\mathbf{u}'_i$ be the appropriately truncated version of $\mathbf{u}_i$ so that $(\mathbf{u}'_i \cdot \mathbf{f}) = (\mathbf{u}_i \cdot \mathbf{g})$.

The vectors $\mathbf{u}'_i$ can be considered to be "analysis" functions and the vectors $\mathbf{v}_i$ can be considered to be "basis" functions for the space of factors (BRDF approximation):

$$\beta_i = \frac{\mathrm{pos}\,(1 - i\,\gamma)}{s_i}(\mathbf{u}'_i \cdot \mathbf{f}), \tag{3.11}$$

$$\mathbf{y}^* = \sum_{i=0}^{N-1} \beta_i \mathbf{v}_i. \tag{3.12}$$

In practice the singular values do not always vanish quickly, which implies that different data sets can emphasize different basis vectors. Nevertheless, multiplying the pseudo-inverse by the sample vector is still faster than performing SVD from scratch and then applying the result to find the BRDF approximation.

Finding the coefficients in Equation 3.11 is analogous to finding the coefficients for the spherical harmonics representation by summing the products of samples and corresponding values of basis functions. However, for spherical harmonics, or any other basis, it would also be necessary to weigh the products with an inverse of the sample density function to get the correct measure for integration. We "bake in" the correct measure when we compute the pseudo-inverse.

Preinversion of the Constraint Matrix

During a composition of the constraint matrix, the entries in its rows are set according to where a given sample is projected. This projection is calculated by the mapping of the light and view directions of a sample to the factor's domain. The projected sample parameters determine, then, the values of the entries in the matrix. Thus, the matrix B (and B^+) depends only on the projection functions π_j and the sampling pattern $(\hat{\omega}_i, \hat{\omega}_o)_i$. It does *not* depend on the sample values; the sample values are used only in a calculation of the factors.

Because the constraint matrix does not depend on the sample data, neither does its pseudo-inverse B^+. Therefore, if we perform the pseudo-inverse calculation once, we can apply the result to many different data sets $\mathbf{f}$, under the constraint that all sets have the same sample directions and the factors use the same projection maps. Calculation of the pseudo-inverse is computationally expensive and is much slower in comparison with applying it to the data vector. Saving the pseudo-inverse and reusing it for other sample sets will drastically reduce the time of the factorization process. In fact, if we have a pseudo-inverse matrix, the calculation of the factors can be performed at interactive rates.

3.2.3 Mapping Between Linear and Log Spaces

To convert a product into a sum, which is required for setting up a system of linear equations, homomorphic factorization takes the logarithm of the equation $f = \prod p_j$. It was noted in the original paper [20] that sample values close to zero get transformed into minus infinities. Those infinities will dominate the equation,

yielding a poor approximation that depicts an extremely absorbing material. This is especially true when the base color (i.e., the diffuse part) is dark, and a narrow specular highlight is bright. Then, due to error in measurement or painting, some of the dark samples may be taken as a part of the highlight, and the few samples that are probing the specular lobe correctly may be overpowered by a couple of stray samples from the dark base color. If that happens, the factorized approximation will have a much dimmer specular highlight than the original material.

At the preinversion stage, there is no general solution to this problem since the described effect stems from the sample values themselves. However, during the calculation of the approximation, the mapping from linear to logarithmic space can be modified to alleviate this problem. A mapping of samples $f' = a + f$, where a is a positive constant, before taking the logarithm,

$$\tilde{f}' \quad = \quad \log(f') = \log(a + f), \tag{3.13}$$

assigns relatively equal significance to all values of samples because the steep portion of the logarithmic function is avoided, since the samples f are always positive. Now, the factors $\mathbf{p}'_j$ found by multiplying the pseudo-inverse by $\tilde{\mathbf{f}}'$ will approximate the values $f' = a + f$ instead of the sample values f. To find the approximation of the BRDF an inverse mapping must be applied:

$$f \quad \approx \quad \hat{f} - a = (\prod p'_j) - a. \tag{3.14}$$

Subtracting one means that f may be negative if the product's result $\hat{f} \in (0, a)$.

In practice this rarely happens providing the samples are consistent.

Although most of the time it is desirable to assign all samples equal influence on the solution, there may be situations when deemphasizing or emphasizing certain values is preferred. To emphasize the base color of a surface, we can apply the usual identity mapping $f' = f$, as has been done in the original paper.

In situations where relatively few samples capture specular highlights and the approximated material is shiny, it may be necessary to use a mapping that puts emphasis on large samples. The mapping $f' = a - f$, where a is the largest sample value, leads to

$$\tilde{f'} \quad = \quad \log(f') = \log(a - f), \tag{3.15}$$

which maps sample values close to a in a linear space into large negative numbers in logarithmic space. These minus infinities will carry a big weight and influence the approximation more than lower sample values. This mapping can be used only if the samples of f are in the interval $[0, a)$. As in the previous case, when computing the factorization, the solution obtained by multiplying the pseudo-inverse of the constraint matrix by the vector $\tilde{\mathbf{f}}'$ yields the solution that is an approximation to transformed samples and not to the samples themselves. An inverse mapping is required to obtain an approximation to the original BRDF:

$$f \quad \approx \quad a - \hat{f} = a - \prod p_j. \tag{3.16}$$

Again a problem with negative BRDF approximations arises when $\hat{f} > a$, but if

samples are coherent, such situations occur rarely. If they do occur, we handle them by clamping to zero, as in the previous case (with some additional error).

The transformation between linear and logratithmic spaces is independent of the pseudo-inverse, because it is performed on the samples before they are multiplied by the matrix. Thus, it is possible to try several mappings quickly with a single pseudo-inverse, and pick the one with the smallest error.

3.3 Factorization on Graphics Hardware

An interesting way to compute the factorization is to utilize graphics hardware [8, 13]. Suppose $\mathbf{u}'_i$ and $\mathbf{f}$ are stored in texture maps. Then, for each i, we can perform a single dot product by rendering a rectangle containing these texture maps, multiplying them together, and then using mip-map generating extensions to add up all the products. Alternatively (and more portably), we could perform several dot products in parallel by storing the j components of all analysis functions together in a single texture map t_j, and using compositing operators to perform the summation over all components.

When done, the necessary coefficients should be stored in a single image, which can be transferred to a texture map. Once such an approximation is computed, a given BRDF (or other function) can be represented by the β_i factors alone (stored in an image with only K pixels), assuming the basis functions $\mathbf{v}_i$ and scale factors $\mathrm{pos}\,(1 - i\,\gamma)/s_i$ are separately known (i.e., are also stored in texture maps). With this approach, the basis functions only need to be stored once regardless of the number of BRDFs stored in the system. For higher performance the summation

of basis functions can be performed (again, possibly using hardware acceleration) during a precomputation phase to compute the necessary factors and store them in texture maps.

3.4 Summary

In this chapter we reviewed the homomorphic factorization as a method for approximating BRDFs with a product of functions of small dimensions. This method can be extended in certain situations by taking advantage of the isotropic nature of many materials that yield radially symmetric 2D factor functions under common parameterizations. By reducing factors from two dimensions to one, we decrease the storage size required by the factors and the constraint matrix. This also results in a reduction of computation time of the pseudo-inverse. The processing time can be reduced even further for compatible sampling patterns by reusing the preinverted matrix. Furthermore, we presented several mappings from linear to logarithmic space, which provide more control over the importance of the different samples. Finally, we have outlined an accelerated technique of factor computation on graphics hardware.

Chapter 4

Recovering BRDFs from Images

Measurement of a specimen's BRDF is usually performed with a gonioreflectometer. This apparatus takes sample measurements of light for hemispherical positions of a source and a detector. Because this procedure is tedious and error-prone, many improvements have been suggested, and, among others, Marschner at al. [17] have proposed a measurement technique that uses photographs to capture many samples of the reflectance function at once. Since the BRDF measurement requires both view and light vectors (beside the light power measurement itself), they calculate them from the geometry acquired with a laser range scanner. The authors use the photograph pixels in conjunction with these vectors to fit the measured samples to a theoretical reflectance model. This results in a BRDF model that approximates a real world material that was measured.

To modify the reflectance model, we can readjust the parameters of a fitted theoretical representation. This, however, can be unintuitive. Also, the BRDF we are looking for is not guaranteed to be in the space of a given representation. An

alternative approach to changing the BRDF is to use a virtual painting application for re-touching the photograph. Changing the appearance of the measured object in the photograph and running the algorithm will produce a modified BRDF. Extending this method further, an entirely new material can be defined by using a painting application to color the object from scratch.

However, one important difference between a painted image and a photograph is that a photograph always depicts a physically realistic reflectance model, while a painting may not do so, due to its non-photorealistic nature. For example, coloring a red plastic sphere will require the user to also paint a bright highlight indicating shininess, and, although the highlight physically should be round and circular, human inaccuracy certainly will cause it to deviate from the correct shape and color. Even though these artifacts occur in the painting, they will not be present in the reflectance model, because during the computation of the reflectance, the sample values, including the outliers, will be averaged out.

This brings us to an important point, that a painting and a rendering are visually separate and different abstractions of an object. This means that an object, rendered using a reflection model that was defined by a painting, will not look exactly like that painting. This has implications for the coloring process, because users need interactive feedback of the impact of the brush strokes on the final form of the reflectance model. The user needs to see the reflectance model, or to be precise, an image rendered using reflectance model resulting from fitting data acquired from a given painting. To secure interactive rates of the feedback, the calculation of the reflectance model itself must be fast because it is needed for the

correct rendering of a reference object.

Most fitting algorithms are computationally expensive and cannot be employed in an interactive application involving thousands of sample points. However, one method that can be used for this purpose is homomorphic factorization if we preinvert the constraint matrix as described in Chapter 3. Thus, incorporating the homomorphic factorization into a painting application will create an interactive BRDF definition tool. This, from an artist's point of view, potentially allows a more natural way for defining reflectance models than mathematical descriptions. Instead of programming a complicated shader, it is conceptually more appealing to color an image and let the program compute the textures and reflectance models from it. We will now describe the details involved in recovering shading models from painted pictures.

4.1 Spatially Uniform BRDFs

Our approach to using painting for the definition of the material is based on a coloring book concept. An artist is given a set of pictures of many views of an object rendered using only black solid silhouette lines, and is asked to apply colors to define the material the object is made of. Such coloring will usually involve painting the base diffuse color and a specular highlight reflection. The painted image will be used as an input to the fast homomorphic factorization, by treating its pixels as BRDF samples.

A pixel can be used as a sample only if there are light and view vectors associated with it. These vectors can be calculated from the scene setup that was used

to create the silhouette outline image. Knowing the positions of the point light source, the object, and the camera, it is possible to calculate the required vectors. For each pixel, a ray is traced and intersected with the object. Having the intersection point, both light and view vectors can be computed. For our parameterization we need also the surface normal and possibly the texture coordinates (see Chapter 3.2.1), and these can be also obtained once the intersection point is known. The pixels in the painted image together with the calculated light and view vectors can now be treated as a BRDF sample set and used as an input to the homomorphic factorization or other approximation scheme.

To define a physically plausible BRDF by painting, the painted color must be in agreement with the light direction defined in the scene. For example, if the light is set up to fall from the left of the scene then the left portion of the objects should be painted brighter than the right side. In other words, the painting must reflect the lighting conditions of the scene. The specular highlight must especially be placed on the object according to the lighting direction of the point light source. To help achieve this consistency, we need to somehow guide the painting process. One way of intuitively indicating the light direction is to present the artist with a reference image of the scene in addition to the outline image. The reference image shows a rendered scene, where the object is assigned some standard material (e.g., Lambertian and Phong), from which it is easy to deduce the approximate light direction. The picture painted by the artist should describe how the object would look with new material properties but strictly under these lighting conditions.

It is also possible to allow the artist the choice of the light direction or set of

views, but before the painting begins, a new constraint matrix must be formed and preinverted to obtain interactive feedback. By setting up a set of standard scenes, we relieve the artist from the burden of constructing and preinverting the matrix. For such scenes, there is no need to recompute the pseudo-inverse because the light and view vectors do not change. To obtain a new BRDF approximation, the artist would invoke fast homomorphic factorization on the painted image to multiply new pixel values by the pseudo-inverse. This multiplication is fast, so the user will see the resulting BRDF right away, and if needed, make corrections to the painting and recompute the BRDF again.

4.2 Spatially Varying BRDFs

Using pixels of a painted image as BRDF samples, described in previous section, constitutes the core idea of several techniques that try to recover more complex reflectance models. In this section we will describe two methods that obtain spatially varying material models from paintings.

4.2.1 Four Factor Product

The homomorphic factorization in the form presented in the original paper [20] assumes spatially invariant BRDFs, where objects appear to be made of uniform materials (e.g., plastic, metal, velvet, etc.). In the painting application, the user has freedom to paint objects in any style, and the samples can be, and usually are, inconsistent. That is, there may be two pixels that have the same sample parameters (light and view vectors) but different color values. Such situations may

be unintentional, in which case the pixel with a different color should be treated as an outlier, if there are many other similarly parametrized samples with consistent colors. On the other hand, the user may intentionally paint spatially varying colors to depict several materials in different regions of one object.

To handle such cases we need to resort to the spatial bidirectional reflectance distribution function (SBRDF) [19] also known as the bidirectional texture function (BTF) [23]. This is a six dimensional function: in addition to the usual four parameters of the standard BRDF it has two variables that parameterize the location on the surface much like texture coordinates. This function describes a relation between the incoming and outgoing light at a particular point on the surface.

If we want to use the homomorphic factorization to factorize an SBRDF, at least one factor in the approximating product must depend on surface parameterization. An obvious choice is to have an additional factor that has a straightforward parameterization taking (u, v) texture coordinates of the surface as input parameters. Thus, the approximation will consist of four factors: $p(\pi_p(\hat{\omega}_i, \hat{\omega}_o))$, $q(\pi_q(\hat{\omega}_i, \hat{\omega}_o))$, $r(\pi_r(\hat{\omega}_i, \hat{\omega}_o))$, and $s(\pi_s(u, v))$. These four factors are calculated in the usual manner using homomorphic factorization from SBRDF samples that have the following projection parameters: light vector, view vector, and texture coordinates.

The intuition behind the role of the fourth factor is the modulation of the BRDF. While factors p, q, and r describe the formation of the color from light reflection, the fourth factor, s, scales this color depending on the surface position. It describes the color change due to the variation in the spatial properties of the object and not the light or view vectors. It turns out that this 4-factor product can indeed

reasonably approximate a slowly varying BRDF.

However, the capability of faithful approximation of the SBRDF by this 4-factor approximation partially depends on the resolution of the fourth factor. The maximum practical resolution is bounded by the limited size of the pseudo-inverse matrix. Since the number of columns is equal to the number of texels in all factors, and the number of rows is equal to the number of samples, the maximum factor resolution depends on the sample number: the more samples, the less resolution of the fourth factor, but for practical computations the resolution of the fourth factor can be at most 32 by 32 texels.

4.2.2 Material Mapping Recovery

Another approach to capturing spatially varying BRDFs is material mapping. In this method, the final color at a given point on the surface is assumed to be the result of a linear combination of several simple BRDFs. We assume a BRDF value at a given point (u, v) on an object surface, and thus any SBRDF sample, can be computed by blending together reflectance functions:

$$\sum_{j=1}^{J} c_j(u,v) f_j(\hat{\omega}_i, \hat{\omega}_o) \tag{4.1}$$

for unknown BRDFs f_j.

An algorithm that strives to recover these BRDFs and their coefficients can be based on an iterative approximation process. When computing a factorized approximation assuming spatial invariance of a BRDF using the 3-factor approach (see Section 4.1), there will be a large residual error for the cases when the painted

reflectance is spatially varying. We may calculate the residual error for each sample,

$$\Delta \;=\; f - \; p(\pi_p(\hat{\omega}_i, \hat{\omega}_o))\; q(\pi_q(\hat{\omega}_i, \hat{\omega}_o))\; r(\pi_r(\hat{\omega}_i, \hat{\omega}_o)), \tag{4.2}$$

and try to fit it with another 3-factor approximation, $\Delta \approx p_1\, q_1\, r_1$. Then, the original function will be more closely approximated by $f \approx p_0\, q_0\, r_0 + p_1\, q_1\, r_1$. This process of calculating the remaining residual and approximating it with another factorization can be continued until the desired level of accuracy is achieved.

Unfortunately, to fit the residual error using homomorphic factorization, the error must be positive. To satisfy this requirement, we can scale the approximating values with a scalar coefficient c to be less than any sample value. Then, the residual errors remain positive and lends itself to repeated homomorphic factorization. The residual error for one sample f_i can be computed by

$$\begin{aligned} \Delta_i &= f_i - c_i\, \hat{f}_i \geq 0 \\ \text{where } c_i &= \begin{cases} f_i / \hat{f}_i & \text{if } f_i < \hat{f}_i \\ 1 & \text{otherwise} \end{cases} \end{aligned} \tag{4.3}$$

To ensure that the residual error is positive for all samples, the coefficient c is set to the smallest multiplier c_i of all samples. When evaluating material maps using linear combinations of approximations $\hat{f}_j$, the scaling value c can be thought of as a blending factor c_j for each factorized approximation of the BRDF:

$$f \;=\; \sum_{j=1}^{J} c_j\, \hat{f}_j. \tag{4.4}$$

So far, we treated the coefficient c_j as a constant for a given 3-factor product, but to speed up the convergence, it is necessary to localize its value for surface regions. So, the blending coefficient is going to be a 2D function dependent on the surface position in accordance with Equation 4.1. This function is represented as a texture, and thus each BRDF sample will fall onto exactly one texel. This way, if a certain factorization closely approximates the samples in one region, it will not be scaled down by a small coefficient that may arise from other regions where this approximation grossly overshoots the sample values. Therefore, the residual will be smaller and it will take less iterations to converge to a good approximation.

It is still possible that several samples are assigned to the same texel. In this case we set the texel coefficient to the smallest coefficient value associated with samples in this texel. This is similar to the global coefficients described earlier, but instead of all samples, we consider only those whose coefficients fall on one texel. Even if a texel accommodates many samples, their coefficients will tend to be of similar magnitude, depending on the size of the surface that texel covers. So the advantages of this approach are not compromised.

Because we are using a 3-factor approximation, the coefficient function $c_j(u, v)$ is not involved in the homomorphic factorization, and therefore we avoid the problem of limited resolution that we encountered in case of the 4-factor approximation (see Section 4.2.1). The coefficient resolution may be arbitrarily large, and high frequency variations can be accurately reproduced, provided the factors p_i, q_i, and r_i can approximate the sample values and the residual error.

If the texture resolution is large, a phenomenon similar to aliasing may occur.

When the 2D blending coefficient for each BRDF is being computed, the samples may fall sparsely on the coefficient's domain. Texels to which no sample is assigned will not be set properly. Their default value will remain one, even though the neighboring texels may have much lower magnitude. During rendering high frequency noise can be seen because consecutive image pixels use alternately high and low coefficient values.

To fix this problem, the values of the texels where the samples fall need to be interpolated. The interpolation will fill the regions of texels inbetween the samples as shown in the Figure 4.1. An easy way to perform this interpolation is to take advantage of rasterization capabilities of graphics hardware and render the sample values using a p-buffer. If we render sample texels as triangles, the texels inbetween these samples will be assigned interpolated values during rasterization.

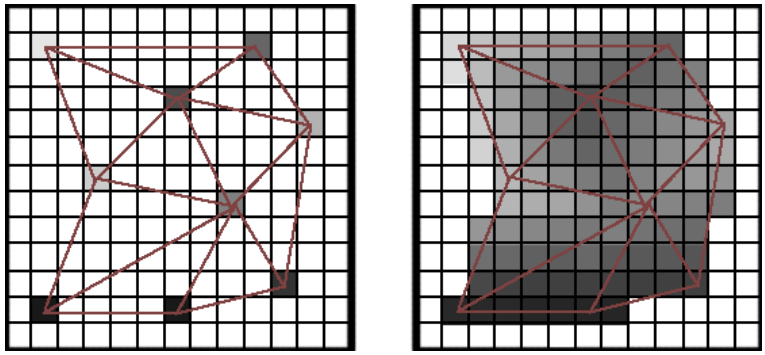


Figure 4.1: Left: coefficient texels set by sparse sampling. Right: unconstrained coefficient texels interpolated through rasterization in texture space.

Before rasterization is used for filling the gaps, the sample points need to be assembled into triangles in the coefficient (i.e., texture coordinate) domain. Constructing triangle meshes in the domain from points may be too expensive for

interactive applications. The triangulation can, however, be precomputed, because sample locations in the coefficient domain are determined by sample locations on the surface. These locations are constant for a fixed sampling pattern, which was assumed in our technique.

We can avoid this precomputation if we further assume coherence of the uv-parameterization on the object's surface. If this condition is satisfied then a mesh in object space is isomorphic to a mesh in the texture space, and therefore a triangle in a model mesh will correspond to a proper triangle in the coefficient domain.

Because we are sampling in a screen space, ultimately we would like to use sample pixels to define a triangle mesh in the coefficient space. First we triangulate the pixels by constructing the triangle strips from regularly sampled pixels. Then, we project this mesh from screen onto the surface and extract the texture coordinates, from which we finally construct the triangular mesh in the coefficient domain.

The mapping from screen to coefficient space has two types of discontinuities. The first one is caused by silhouette edges, where two neighboring pixels will be transformed to two distant points in the target space. The second one is caused by discontinuity in uv parameterization where two neighboring points on the surface have very different texture coordinates usually due to the texture wrapping. It is important, then, that the mesh triangles in image space do not cross the silhouette boundaries of the rendered object. Thus, we need to test for silhouettes and uv-cuts during the triangulation of the sample points in image space.

4.3 Hybrid Method

The material mapping technique presented in the previous section is using standard 3-factor approximations as representations for subsequent BRDFs that compose a given material. However, the material mapping and the 4-factor approximation methods are not mutually exclusive, and they can be combined, in which case material mapping would use 4-factor instead of 3-factor approximations. Apart from that change, the procedure of calculating such a representation is unaltered. The resulting SBRDF approximation takes slightly more space since 2D factors are used in addition to the usual 1D factors, but the solution tends to converge faster on the desired solution.

4.4 Integration of Multiple Views

When defining spatially varying BRDFs on an object, it is important to paint the whole surface to define material variation everywhere. Otherwise, during rendering or retargeting, the portions of an object that were lacking samples will not have a defined spatial reflectance behavior.

One painted view is not enough to visually encompass all portions of the object. We need to use several paintings of the illuminated model capturing views from different directions. Integration of the information contained in each view may be challenging especially if the paintings on different views are inconsistent with one another. However, assuming some degree of coherence, it is possible to find factorized approximations using several views.

To find a 3- or 4-factor approximation, each valid pixel in each view will participate in setting up the constraint matrix. The views can differ in lighting conditions (i.e., light position) but each painting must be consistent with its own lighting. Setting up the constraint matrix is conducted in the usual way: for each pixel the projected parameters are used to set up entries in the matrix. Finding the factorization using the pseudo-inverse is also the same. The pixel values are multiplied by the pseudo-inverse to obtain the factors.

Recovering a material map from several views is more tricky. In each iteration of the algorithm we need to ensure that the residuals are always positive. Thus, during one iteration for each view we calculate the coefficient texture that makes the residual positive for this particular view. Then we merge the coefficient textures, keeping always the smallest value. Keeping the smallest value ensures that the contributing BRDF is scaled down enough to be less than any sample from any view.

4.5 Summary

This chapter has presented our techniques that recover reflectance and shading models from painted images. All methods use pixels as BRDF samples. These are used as an input to the fast homomorphic factorization algorithm to approximate a reflectance model. Spatially varying BRDFs can also be approximated. Two techniques, a 4-factor and a material mapping described in this chapter, reconstruct the SBRDF. The 4-factor method introduces an additional factor to capture spatial variations, while the material mapping blends several uniform BRDFs using

spatially dependent coefficients.

For all techniques multiple images of the object are used. Each of the images defines a 2D slice through a 4D BRDF or a 6D SBRDF, and from this slice the whole function is approximated. Each valid pixel in every image is treated as a sample of the function, and is used for approximation.

These techniques can be used for definition of non-conventional reflectance and shading models. Spatial reflectance functions that can be approximated by the 4-factor method are limited to low frequencies. The material mapping does not have this drawback, but many more views must be used with consistent spatial painting to achieve good results. However, the real strength of all the techniques lies in the speed allowing their use in an interactive application.

Chapter 5

Results

This chapter provides time measurements that illustrate the performance improvement of fast homomorphic factorization. Several pictures are also presented for comparison of the painted input images and resulting shading models. We start by analyzing the algorithm's performance.

Table 5.1 shows time measurements of SVD computations and factorizations using 7884 samples. The calculation times for 4-factor approximation are much larger than for 3-factor factorization because of the number of texels that the fourth 2D factor introduces into the approximation. In all cases, however, the factorization time using preinverted constraint matrices are orders of magnitude smaller than performing the calculations from scratch. Essentially, this table illustrates time saved by preinverting and storing the pseudo-inverse of the constraint matrix. By reapplying this matrix, we avoid the SVD recomputation time shown in the two left columns of the Table 5.1.

Table 5.2 shows similar measurements for approximations that used 2D func-

tions (instead of 1D as in Table 5.1). Again, the computation time is much smaller in the fast homomorphic factorization cases, especially for 3-factor approximation, where three 2D factors result in a large constraint matrix inversion. The sample count had to be reduced to around 300 to bring the constraint matrix down to a manageable size. The use of 2D factor functions is necessary for anisotropic materials (see Figure 2.2). Although the speed improvement is most evident for the anisotropic cases, the applicability of the fast homomorphic factorization is undermined by the small number of samples imposed by the matrix size.

res.	SVD computation		fast factorization	
	3-factor	4-factor	3-factor	4-factor
32	4.26	247.36	0.1834	1.2649
64	11.08	330.04	0.2868	1.3207
96	21.58	397.37	0.3842	1.5919
128	38.06	506.11	0.4375	1.6877

Table 5.1: SVD computation times in seconds for preinversion and factorization (1D factors) using 7884 samples. The resolution of the fourth function in the 4-factor product, which is 2D, is constant and equal to 32x32 texels. Resolution changes only for 1D factors. The number of rows in the constraint matrix for 3-factor approximation is $3 \times res$ and for 4-factor approximation is $3 \times res + 32^2$. The number of columns is 7884.

Table 5.3 shows the time measurements for material map recovery using 3-factor BRDF approximation for 7884 samples. The iterative calculations increased the calculation time, but the speed remained interactive. Although more BRDFs yield a better approximation, in practice, around five BRDFs is sufficient. This table (Table 5.3) is primarily intended to show the computation time trend associated with increasing number of approximated BRDF components.

The approximation error incurred by the approximation depends on the sample

res.	SVD computation		fast factorization	
	3-factor	4-factor	3-factor	4-factor
16	1m 43s	3.77s	0.0308s	0.0160s
32	101m 12s	13.76s	0.1048s	0.0418s
48	-	44.88s	-	0.0853s
64	-	121.55s	-	0.1409s

Table 5.2: SVD computation times in seconds for preinversion and factorization (2D factors) using 317 samples. In 3-factor products all three functions are 2D, and in 4-actor products the first three functions are 1D (kept at 32 texels) and only the fourth factor is 2D. Resolution changes only for 2D factors. The number of rows in the constraint matrix for 3-factor approximation is $3 \times res^2$ and for 4-factor approximation is $3 \times 32 + res^2$. The number of columns is 317.

res.	Num. of BRDFs			
	5	10	15	20
32	2.44	5.73	8.91	12.09
128	3.96	8.66	12.64	17.79

Table 5.3: Material map computation time in seconds.

coherency. This is most evident in the spatially varying paintings that are approximated using 4-factor factorization or material map recovery techniques. The error between the approximated and painted triceratops model is shown in Table 5.4 and corresponds to figures Figures 5.5(b) and (c).

	4-factor	Material Map (Num. of products)				
		1	2	3	4	5
RMS	0.15	0.18	0.11	0.08	0.06	0.05
max	4.32	0.83	0.56	0.49	0.46	0.43

Table 5.4: Error metrics for 4-factor approximation and material map approximations using between 1 and 5 product components. (Sample values are in the interval $[0, 1]$.)

Figure 5.1 shows a rendering of a cow using standard OpenGL (left) and using a factorized approximation of that lighting (right). It is interesting to see that 3-

factor approximation captures both diffuse and specular reflection of light, although the Phong model cannot be mathematically factorized into a product.

The standard parameterization of factors is capable of handling non-photorealistic effects as shown in Figure 5.2. An outline of a duck is painted and then captured by factorization. This method, however, suffers from a well known problem of silhouette thinning for objects with high curvature (e.g., the duck's beak).

Figure 5.3 illustrates the effect of smoothing on the solution. Smoothing is intended to interpolate the gaps where there is no samples. If sufficiently many samples are taken, the influence of smoothing term is minimal. In this example, 207 samples were taken, which is an order of magnitude less than in other examples. Figure 5.3 (b) shows the solution with no smoothing constraints that resulted in Gibbs phenomenon. Images (c) and (d) of this figure show the effect of increasing the value of the smoothing coefficient.

Figure 5.4 shows a coarse painting of a highlight on a diffuse sphere, and re-targeting this reflection model to render a cat. The placement of the highlight on the sphere must be accurate, otherwise visible dark and bright rings appear as artifacts. This is because the highlight recovery is sensitive to the projection of the high intensity samples onto incorrect texels. This effect can be reduced by applying one of the alternative mappings from linear to log space.

Figure 5.5 compares the reconstruction of the painted picture (a) by material mapping (b) and 4-factor approximation (c). The material mapping uses five BRDFs each approximated by 3-factor factorization. The material mapping case approximates the samples better (compare Figure 5.5(b) and (c) to original in (a)),

but a 4-factor approximation gives in practice much better visual quality when viewed from other angles (compare Figure 5.5(e) to (f)). The best results when using only one painting were achieved by the hybrid method that used material mapping technique where each BRDF component was approximated with a 4-factor factorization (Figure 5.5(d)). To match this visual quality when using material mapping with standard 3-factor approximation (i.e., to improve Figure 5.5(e)), paintings from several view directions are required as shown in Figure 5.6.

The material mapping approach yields better results if the painting style is close to physical reality. Figure 5.6 shows reflectance recovery from multiple views using material mapping. Three images of one object are painted from different viewing directions and the fourth image shows a new rendered image from another direction.

Possible results obtained by the technique are illustrated in figure Figure 5.8. Top rows show painted models and the bottom row shows the recovered shading models, BRDFs, and SBRDFs. The sphere in Figure 5.8(a) is painted in a golden color with a bright highlight. It is a representation of Phong lighting model. An artistically painted highlight is shown in Figure 5.8(b), where a yellow specular reflection has a green and blue halo. Figure 5.8(c) illustrates a reproduction of a velvet material. The input for this picture was an image processed photograph of a ball shown in Figure 5.7. A non-photorealistic rendering example is shown in Figure 5.8(d) with warm to cool shading. The warm yellow color is applied for surface facing light and cool blue shade for surface facing away. A spatially variant diffuse color is captured in Figure 5.8(e), and spatially variant BRDF in Figure 5.8(f). Note that in Figure 5.8(f) the specular highlight occurs only in one

type of wood fiber while the other one is more diffuse.

5.1 Discussion

This section discusses issues accompanying the methods described in this thesis. We start with the fast factorization and then proceed to the recovery of BRDFs.

The gain in factorization speed is achieved by sacrificing flexibility of sampling direction choice. Thus, the only applications that can take advantage of this improvement are the ones for which the sampling pattern is fixed. If the light or view direction changes from previous measurement settings, the constraint matrix needs to be rebuilt and its pseudo-inverse recomputed. Fortunately, the limitation to fixed sampling pattern is not always an issue, as, for example, in the painting tool case.

Even when the fixed sampling pattern warrants the use of the fast homomorphic factorization, new issues may still arise. The fast factorization uses a preinverted matrix whose size ranges from 10MB to 100MB, depending on the number of samples and factors. For fastest performance, this matrix should be loaded in main memory, which may degrade performance on computers that have an inadequate amount of RAM. In such cases, the algorithm can be modified to avoid the loading of the whole pseudo-inverse to the main memory, but it will suffer the delay penalty of data transfer from a hard drive during the computation process. In such circumstances, a sparse QR factorization may be useful, because it can improve space requirements of the QR-factorized constraint matrix that is equivalent to its pseudo-inverse.

As described above, the size of the pseudo-inverse matrix and the fixed pattern requirement are two major issues associated with fast homomorphic factorization. In addition there are also some issues regarding the painting and BRDF recovery.

One of the issues pertains to a specular highlight, which needs to be painted for correct definition of many reflectance functions. The fast factorization is sensitive to the placement of such highlights. If the highlight is centered incorrectly, the approximation will suffer from ringing (Figure 5.3) artifact showing as alternating pattern of darker and lighter rings. This can be reduced by applying the new mapping from linear to logarithmic space presented in the thesis. However, the only way to fully alleviate the rings is by iterative corrections of the painting itself.

Despite the potential usefulness of the application, achieving desired reflectance model is not always easy. The problem arises from the conflict between real world shading parameterization and the non-photorealisic nature of painting. A best fit solution solves some of the problems caused by this decoupling. If most of the pixels are close to being in agreement with reality, the average is going to approximate the reflectance model well enough. However, if there are many discrepancies in the painting, a less desirable solution is produced. Again, an iterative painting process needs to be employed to converge on a satisfying solution. This can be time consuming and does not even guarantee that the desired shading model can be approximated adequately.

For spatially varying materials, part of the reason why an approximation may not capture the intended shading model is an ambiguity between variation in a BRDF and a material itself. For example, a localized increase in brightness on

an object can be attributed either to the specular reflection or a brighter color (albedo) of the material at that location. There is no clear solution other than painting several other images from different views or with different light directions to disambiguate the source of the brightness.

Our techniques for capturing spatially varying materials are admittedly ad hoc and have some limitations. For example, there is no guarantee that the residual error in material mapping technique can be factorized, since it may not resemble any form of reflectance function. Again, there is a trend that paintings that diverge from physical plausibility will cause more problems than paintings having physical reflectance properties. This is illustrated in Figure 5.6 and Figure 5.5(e), where marble is adequately captured by material mapping while non-photorealistic painting is not. As before, to pinpoint the correct shading model, more images are required from different view and for different lighting conditions.

A trend can be observed that for an adequate definition of a BRDF, paintings from many views or light directions are required. This is because we are striving to define 4D or 6D functions (for BRDFs and SBRDFs respectively) from 2D images. One image can be thought as a 2D slice through a function of higher dimensions, and so many slices are needed to define it. However, the images have to be consistent, and in particular the spatial variations must "stick" to the surface and their patterns must appear in the same places on the object in all of the images. This task is difficult for hand drawn paintings, but required, because the factorization is sensitive to the mismatched spatial patterns.

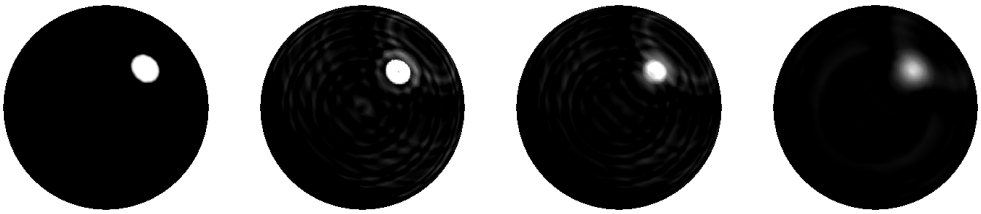
Lensch et al. [14], McAllister et al. [19], and Hertzman et al. [7] have accurately

recovered spatially varying materials from photographs by fitting sample data to a Lafortune's reflectance model. Such approach yields superior results, but requires a solution to a non-linear optimization problem. Nonlinear solvers, based on the conjugate gradient method, take hours to compute the result, and, thus, are not suited for interactive applications. Our technique, on the other hand, is fast and permits iterations to remove inconsistencies in the paintings.

The number of samples involved in the computations could contribute to the speed difference between our and their algorithms. Our system uses tens of thousands sample points, while the other techniques use millions of samples. It would be interesting to investigate the application of the mentioned techniques to BRDF recovery from paintings.

Figure 5.1: Diffuse and Phong specular reflectance model. Left image was rendered analytically and right one was rendered using factorization. Due to smoothing, the highlight on captured model is not as sharp as it is on the analytical model.

Figure 5.2: Painted and rendered duck model. Please note that smoothing gives blurred edge.



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

Figure 5.3: The effect of smoothing coefficient on the solution: (a) the input image and the approximations with coefficient values of 0.0 for (b), 1.0 for (c), 10.0 for (d).

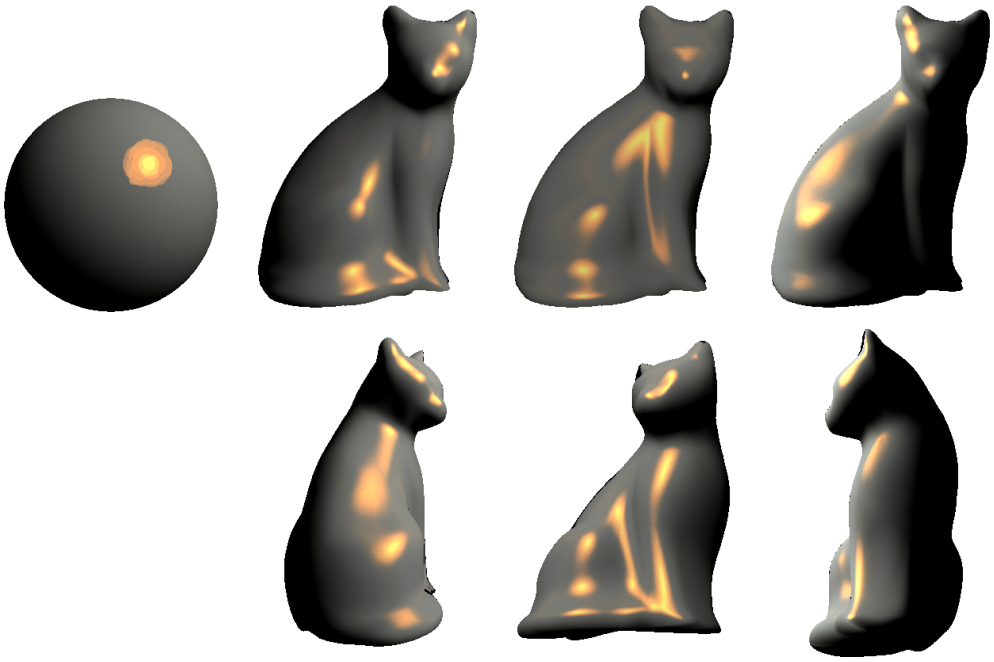
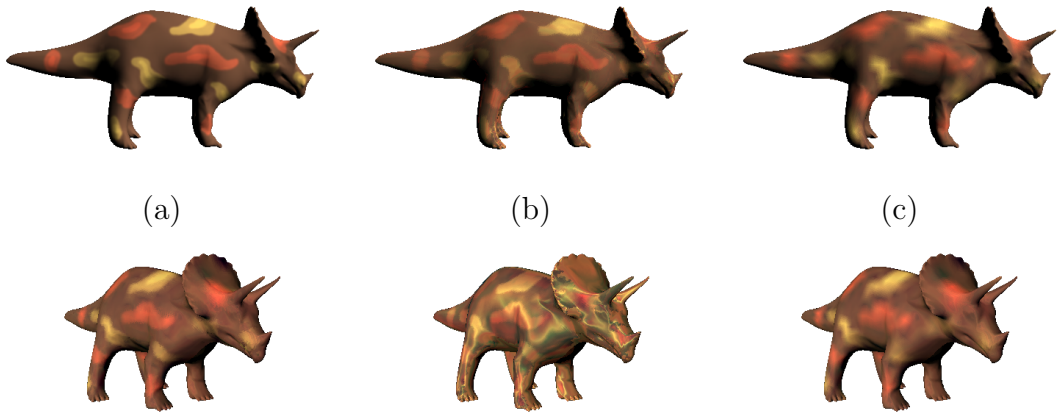


Figure 5.4: A painted sphere and a cat under changing light and view direction.



(a) (b) (c)

(d) (e) (f)

Figure 5.5: Material captured from one painting: (a) painted triceratops (b) material map (c) 4-factor approximation. Material rendered under new view and lighting directions: (d) hybrid of material map and 4-factor approximation (e) just material map (f) just 4-factor factorization.

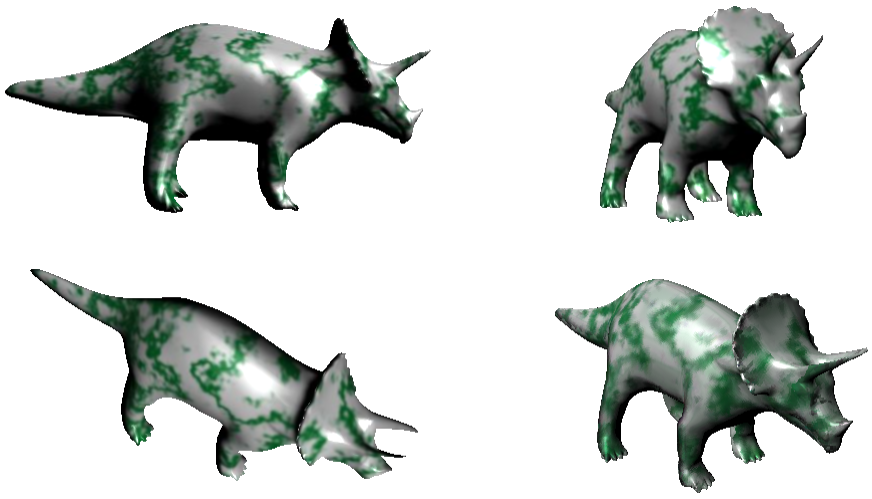


Figure 5.6: Three views of a triceratops painted with marble BRDFs and in bottom right a triceratops under new lighting direction viewed from new angle. The reflectance model is recovered using material mapping with five BRDFs.

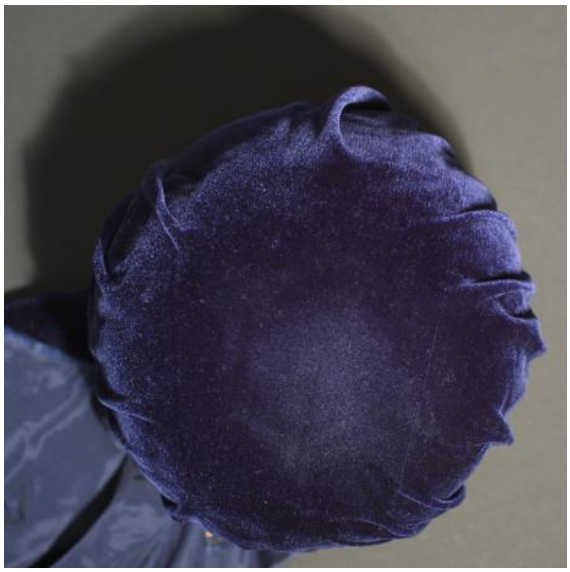


Figure 5.7: A photograph of a sphere wrapped in a velvet material.

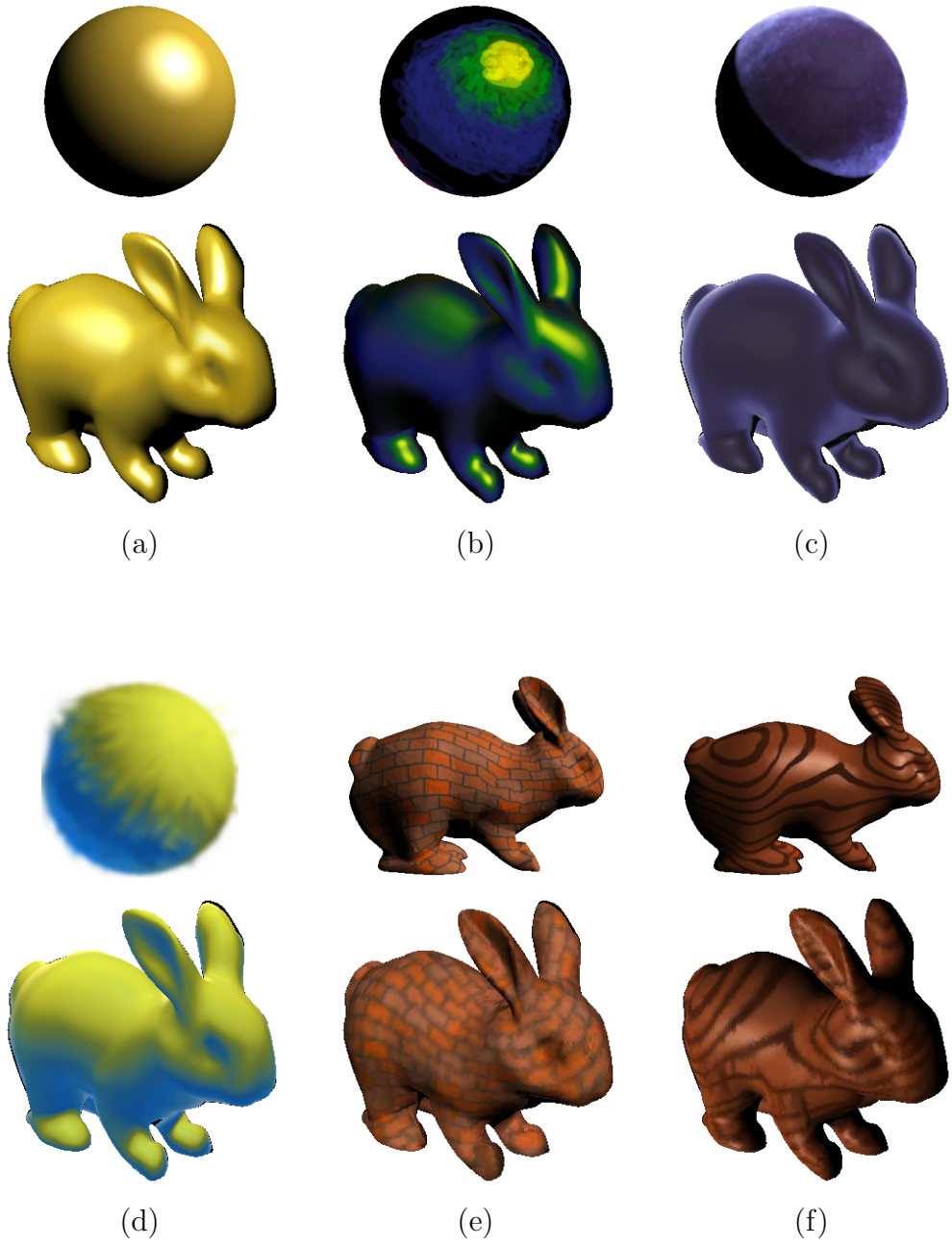


Figure 5.8: Painted and recovered shading models: (a) golden (b) yellow highlight with green and blue halos (c) velvet (d) warm to cool (e) diffuse brick (f) wood consisting of two BRDFs.

Chapter 6

Conclusion

This thesis has presented a refinement of the homomorphic factorization and its application to the definition of BRDFs and SBRDFs from painted images.

The speed of the new fast homomorphic factorization algorithm stems from the precalculation of the pseudo-inverse and the ability of its application to any samples that have similar pattern and layout as the samples used to define a constraint matrix. Additional speed improvement is gained by replacing 2D factors with 1D factors for the special case of isotropic materials.

The definition of BRDFs is achieved by painting examples of objects depicting the intended reflectance model. The image pixels are then treated as an input to the fast homomorphic factorization obtaining the shading model. The spatially variant BRDFs are recovered by either a 4-factor product, a material mapping technique, or a hybrid of both. Also, new mappings are introduced to allow for control over the influence of low sample values, which tend to have a greater impact on the final approximation than high values if a standard mapping is used from linear to

logarithmic space.

The overall accuracy of BRDF approximation depends on the painting style. Since the factorization is based on physical attributes, its strength lies in recovering physically plausible reflectance models. Nevertheless, non-photorealistic effects were successfully captured using this algorithm.

6.1 Future Work

Both fast homomorphic factorization and the painting tool can be further improved. For example the the factorization could be extended to allow any sampling directions. This would increase the flexibility of the factorization technique while preserving the speed. It may be possible that the samples could be projected against different analysis vectors (see Section 3.2.2) that reflect new sampling pattern. Thus, factors could be calculated for changing sampling patterns without the recomputation of the pseudo-inverse, which would increase the applicability of the fast homomorphic factorization. This would also open the possibility of using other basis functions, using both linear and nonlinear fitting models under user direction.

Incorporating anisotropy is another possible extension. Representing such materials is already supported if we keep the factors two dimensional. Because factorization resulting in 2D factors is time consuming (due to the number of table elements to constrain), the fast factorizing algorithm is going to be especially valuable. The painted object would need to have local frames mapped on its surface to define the preferred direction of anisotropy, and the main obstacle would be making the paintings consistent with this direction.

In our work we focused on the homomorphic approximation technique for its novelty value. However, the same approach of using painted values as BRDF samples could be used with any approximation technique, such as spherical harmonics, polynomial texture mapping, or Lafortune lobes, or a hybrid of these techniques, if they could be made fast enough. The SVD could be also replaced by a sparse QR factorization, which is potentially more efficient. QR factorization may also offer a more compact representation of the pseudo-inverse, thus allowing more samples and larger resolution of approximating factors.

It would also be straightforward to generalize the approach to approximation of outgoing radiance values rather than just reflectance models under point illumination. The paintings would have to take into account the environment instead of a single light source. The paintings would capture the view from different directions and for different orientations of the object with respect to the surrounding environment (i.e., in the world coordinate system).

In addition to these extensions, it would be interesting to investigate other parameterizations that would be more suitable for non-photorealistic imagery.

The clustering of samples that represent the same material has been shown to improve the quality of captured SBRDFs [14, 7]. These methods could potentially be applied to homomorphic factorization if a sample subset could be selected for participation in factorization. This may be achieved by assigning weights to samples on both sides of Equation 3.3. These weights can be then factored out of the constraint matrix as a diagonal matrix of weights that is easily invertible. During the factorization process, the weights can be set to zero for the samples that do not

belong in the cluster of a particular material. This approach, however, raises some questions and needs to be further researched. For instance, inverting zero weights that are present in the diagonal matrix is undefined.

Other future work could involve following up on previous findings of Lensch [14], McAllister [19], and Hertzmann [7] to improve the performance of these algorithms. Recent advancements presented by Hillesland et al. [8] have reduced the computation time of non-linear optimizers fivefold by exploiting graphics hardware. Despite these results, the performance is still far from being interactive, and more work needs to be done in this area.

Bibliography

[1] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional Reflection Functions From Surface Bump Maps. In *Proceedings of SIGGRAPH*, 1987.

[2] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the Reflectance Field of a Human Face. In *Proceedings of SIGGRAPH*, 2000.

[3] Alain Fournier. Separating reflection functions for linear radiosity. In *Eurographics Rendering Workshop*, 1995.

[4] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter S. Shirley, and Rich Riesenfeld. Interactive Technical Illustration. In *Symposium on Interactive 3D Graphics*, 1999.

[5] Pat Hanrahan and Paul E. Haeberli. Direct WYSIWYG Painting and Texturing on 3D Shapes. In *Proceedings of SIGGRAPH*, 1990.

[6] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, Hardware-Accelerated Shading and Lighting. In *Proceedings of SIGGRAPH*, 1999.

[7] Aaron Hertzmann and Steven M. Seitz. Shape and Materials by Example: A Photometric Stereo Approach. In *In Proceedings of IEEE CVPR*, 2003. To appear.

[8] Karl E. Hillesland, Sergey Molinov, and Radek Grzeszczuk. Nonlinear Optimization Framework for Image-Based Modeling on Programmable Graphics Hardware. In *SIGGRAPH*, 2003. To appear.

[9] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. WYSIWYG NPR: Drawing Strokes Directly on 3D Models. In *Proceedings of SIGGRAPH*, 2002.

[10] Jan Kautz and Michael D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Eurographics Rendering Workshop*, 1999.

[11] Jan Kautz and Hans-Peter Seidel. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. In *SIGGRAPH / Eurographics Workshop on Graphics Hardware*, 2000.

[12] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-Linear Approximation of Reflectance Functions. In *Proceedings of SIGGRAPH*, 1997.

[13] E. Scott Larsen and David McAllister. Fast Matrix Multiplies Using Graphics Hardware. In *ACM/IEEE Conference on Supercomputing*, 2001.

[14] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-Based Reconstruction of Spatially Varying Materials. In *Rendering Techniques: Eurographics Rendering Workshop*, 2001.

[15] A. A. Maciejewski and C. A. Klein. The Singular Value Decomposition: Computation and Application to Robotics. In *International Journal of Robotics Research*, 1989.

[16] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial Texture Maps. In *Proceedings of SIGGRAPH*, 2001.

[17] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-based BRDF Measurement Including Human Skin. In *Eurographics Rendering Workshop*, 1999.

[18] Pontus Matstoms. Sparse QR Factorization in MATLAB. In *Transactions on Mathematical Software*, 1994.

[19] David McAllister, Anselmo Lastra, and Wolfgang Heidrich. Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions. In *SIGGRAPH / Eurographics Workshop on Graphics Hardware*, 2002.

[20] Michael D. McCool, Jason Ang, and Anis Ahmad. Homomorphic Factorization of BRDFs for High-Performance Rendering. In *Proceedings of SIGGRAPH*, 2001.

[21] Ravi Ramamoorthi and Pat Hanrahan. A Signal-Processing Framework for Inverse Rendering. In *Proceedings of SIGGRAPH*, 2001.

[22] Peter-Pike Sloan, William Martin, Amy Gooch, and Bruce Gooch. The Lit Sphere: A Model for Capturing NPR Shading from Art. In *Graphics Interface*, 2001.

[23] Xin Tong, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo, and Heung-Yeung Shum. Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces. In *Proceedings of SIGGRAPH*, 2002.

[24] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting Reflectance Functions From Complex Surfaces. In *Proceedings of SIGGRAPH*, 1992.