Chapter 5

Frequency Space Environment Map Rendering

In the previous chapter on irradiance environment maps, we have seen how frequency-space analysis can be used to efficiently compute and represent convolutions of the incident illumination with the low-pass Lambertian reflection function, i.e. clamped cosine. An obvious question is whether a similar approach is possible for general BRDFs. In this chapter, we will present such a method. We will therefore have shown that the signal-processing ideas can be used to significantly increase the quality of interactive rendering. Since we are using frequency space methods to render with environment maps, we have entitled this chapter *Frequency Space Environment Map Rendering*. The next chapter of the dissertation will switch from forward to inverse rendering, extending and applying the theoretical framework to determining lighting and BRDFs from sequences of real photographs.

This chapter presents a new method for real-time rendering of objects with complex isotropic BRDFs under distant natural illumination, as specified by an environment map. Our approach is based on spherical frequency space analysis and includes three significant improvements over previous work. Firstly, we are able to theoretically analyze required sampling rates and resolutions, which have traditionally been determined in an ad-hoc manner. We also introduce a new compact representation, which we call a *spherical harmonic reflection map (SHRM)*, for efficient representation and rendering. Finally, we show how to

Figure 5.1: *These images, showing many different lighting conditions and BRDFs, were each rendered at approximately 30 frames per second using our Spherical Harmonic Reflection Map (SHRM) representation. From left to right, a simplified microfacet BRDF, krylon blue (using McCool et al.'s reconstruction from measurements at Cornell), orange and velvet (CURET database), and an anisotropic BRDF (based on the Kajiya-Kay model). The environment maps are the Grace Cathedral, St. Peter's Basilica, the Uffizi gallery, and a Eucalyptus grove, courtesy Paul Debevec. The armadillo model is from Venkat Krishnamurthy.*

rapidly prefilter the environment map to compute the *SHRM*—our frequency domain prefiltering algorithm is generally orders of magnitude faster than previous angular (spatial) domain approaches.

The rest of this chapter is organized as follows. We introduce the problem in section 1, followed by a discussion of previous work in section 2. Section 3 introduces the preliminaries and background required. Section 4 describes our new *SHRM* representation, section 5 discusses an analysis of sampling rates in order to set frequency space orders and resolutions, and section 6 describes our efficient frequency domain prefiltering algorithms. Section 7 presents our results using a number of different environment maps and BRDFs. Finally, we conclude this chapter and discuss future work in section 8.

5.1 Introduction

Our goals are real-time rendering with complex natural illumination and realistic, possibly measured, BRDFs. The closest previous work is that of Cabral et al. [8], who extended standard environment maps by interactively warping and combining a sparse 2D set of prerendered images. These precomputed images were obtained by *prefiltering* the environment map, i.e. integrating the product of the BRDF and lighting over the visible (upper) hemisphere for each image pixel, with each pixel corresponding to a particular surface normal direction. Subsequently, Kautz et al. [38, 40] proposed alternative implementations and improved prefiltering methods.

This paper introduces a new frequency space paradigm for prefiltering and rendering environment mapped images with general isotropic BRDFs. We show that frequency space analysis allows for setting sampling rates accurately, and enables compact representations. Further, just as image convolutions are often computed in the Fourier rather than the spatial domain, prefiltering is more efficient in frequency rather than angular space. Our main contributions are:

Theoretical analysis of sampling rates and resolutions: Most previous work has determined reflection map resolutions, or the number of reflection maps required, in an ad-hoc manner. By using a signal-processing framework, we are able to perform error analysis, that allows us to set sampling rates and resolutions accurately.

Efficient representation and rendering with Spherical Harmonic Reflection Maps: We introduce *spherical harmonic reflection maps (SHRMs)* as a compact representation. Instead of a single color, each pixel stores coefficients of a spherical harmonic expansion encoding view-dependence of the reflection map. An observation that emerges from the theoretical analysis is that for almost all BRDFs, a very low order spherical harmonic expansion suffices. Thus, SHRMs can be evaluated in real-time for rendering. Further, they are significantly more compact and accurate than previous methods [8, 38] that use an explicit 1D or 2D set of images.

Fast prefiltering: One of the drawbacks of current environment mapping techniques is the significant computational time required for prefiltering, which can run into hours, and preclude the use of these approaches in applications involving lighting and material design, or dynamic lighting. We introduce new prefiltering methods based on spherical harmonic transforms, and show both empirically, and analytically by computational complexity analysis, that our algorithms are orders of magnitude faster than previous work.

We present a complete theoretical analysis and practical algorithm pipeline, incorporating all three contributions. It is also possible to separately (incrementally) incorporate any one (or two) of the improvements into previous methods.

5.2 Related Work

Angular space environment map rendering has a long history in graphics, including early work by Blinn and Newell [5], Miller and Hoffman [59], and Greene [22]. Hakura et al. [28] propose location and geometry-dependent environment maps for local reflections. Our goals are different in that we want to capture the effects of complex BRDFs and use any object geometry, but it should be possible in future to combine the methods for local reflections and complex BRDFs.

As noted by Cabral et al. [8], environment mapping can be viewed as reflection-space image-based rendering, and is therefore related to a number of IBR methods like surface light fields [60, 64, 88]. A surface light field stores the outgoing radiance distribution for each point on a geometric model. Our representation is essentially equivalent to the surface light field of a sphere. We store the reflected radiance distribution for each normal direction, allowing our representation to be mapped on to any object geometry.

Our work also relates to recent research on hardware rendering with factored BRDFs [37, 57]. However, these methods require the BRDF to at least approximately satisfy a particular factored form. These previous methods also do not support complex illumination.

5.3 Preliminaries

In this section, we first discuss the reflection equation and introduce the basic framework for our method. We then describe our BRDF parameterization, and discuss previous 4D function representations. We use the same notation as in the rest of the dissertation, summarized in table 1.1. In addition, there is some new notation used in this chapter, summarized in table 5.1.

Assumptions: We make a number of simplifying assumptions common in real-time rendering in general, and environment mapping [8, 38, 40, 59] in particular. Specifically, we assume distant illumination and isotropic BRDFs, and restrict ourselves to direct lighting, ignoring interreflection and self-shadowing. We will also not explicitly considered textured objects, but it is easy to use texture-mapping to modulate the net reflectance, simply

$\hat{B}(\alpha, \beta, \theta_o, \phi_o)$	Reflected light field in terms of fixed global viewing direction
F	Maximum order l of coefficients $\hat{\rho}_{lpq}$,B_{lmpq}
P	Maximum order p in spherical harmonic expansion
S	Angular resolution $(S > F)$
T	Number of images in angular space $(T > P)$
W	Angular width of BRDF
ϵ	Error (unaccounted energy in approximation)
C_a	Angular space computational cost
C_f	Frequency domain computational cost

Table 5.1: *Notation used in chapter 5.*

by multiplying the texture and the reflected light field computed by our methods. It should be noted that it is not easy to modulate BRDF parameters, only the net reflectance.

Reflection Equation: Given our assumptions, we may drop spatial dependence, and write the standard reflection equation (c.f. equation 2.9),

$$B(\alpha, \beta, \gamma, \theta'_o) = \int_{\Omega'_i} L(\theta_i, \phi_i)\hat{\rho}(\theta'_i, \phi'_i, \theta'_o, \phi'_o)\, d\omega'_i. \tag{5.1}$$

As usual, L is the incident radiance, $\hat{\rho}$ is the BRDF transfer function, and B is the reflected light field. (α, β) correspond to the *global* coordinates of the surface normal, (θ_i, ϕ_i) are the *global* coordinates of the incident light direction, and (θ'_i, ϕ'_i) and (θ'_o, ϕ'_o) are *local* incident and outgoing directions. Applying the standard rotation to align local and global coordinates, we obtain (c.f. equation 2.11),

$$B(\alpha, \beta, \theta'_o, \phi'_o) = \int_{\Omega'_i} L\left(R_{\alpha,\beta}(\theta'_i, \phi'_i)\right) \hat{\rho}(\theta'_i, \phi'_i, \theta'_o, \phi'_o)\, d\omega'_i. \tag{5.2}$$

When the viewer is distant, it is often useful to rewrite the reflected light field in terms of the global viewing direction $\vec{V} = (\theta_o, \phi_o)$. Note that so far in this dissertation, we have considered local outgoing angles only, so this is a new notation. To distinguish it from the familiar notation in terms of local angles, we denote the reflected light field, parameterized by global outgoing direction as $\hat{B}$. We now have,

$$
\begin{aligned}
(\theta_o', \phi_o') &= R^{-1}_{\alpha,\beta}(\theta_o, \phi_o) \\
\hat{B}(\alpha, \beta; \theta_o, \phi_o) &= B(\alpha, \beta, \theta_o', \phi_o') \\
&= B\left(\alpha, \beta; R^{-1}_{\alpha,\beta}(\theta_o, \phi_o)\right). \qquad (5.3)
\end{aligned}
$$

Our general approach (and that of previous work [8, 38]) is to represent the incident lighting L by an environment map. The environment map is *prefiltered* to compute some representation of B (or $\tilde{B}$), followed by interactive *rendering* with this representation.

The rest of this chapter covers a number of issues that must be addressed. First, we must find the appropriate (re)parameterization for B and $\hat{\rho}$. Next, we must determine how to represent B in a compact manner suitable for interactive rendering. For this, it is important to analyze the required sampling rates and output resolutions. Finally, we must determine how to efficiently compute our representation of B, i.e. rapidly prefilter the environment map. An overview of our entire algorithm pipeline is shown in figure 5.2.

5.3.1 Reparameterization by central BRDF direction

Our goal is to reparameterize the BRDF and reflected light field so that they become relatively simple and compact, and possibly lower-dimensional functions. Reparameterizing also allows us to eliminate the warping step required by Cabral et al. [8]. To do this, we reparameterize by the central BRDF direction, commonly the reflection vector, as per the discussion in section 2.3.4. We repeat below the important points and discuss some relevant details. Please refer to figure 2.4 in chapter 2 for an illustration of the ideas.

Consider first the special case of *radially symmetric or 1D BRDFs*, where the BRDF consists of a single symmetric lobe of fixed shape, whose orientation depends only on a well-defined central direction $\vec{C}$. In other words, the BRDF is given by a 1D function u as $\hat{\rho} = u(\vec{C} \cdot \vec{L})$. Examples are Lambertian $\hat{\rho} = \vec{N} \cdot \vec{L}$ and Phong $\hat{\rho} = (\vec{R} \cdot \vec{L})^s$ models. If we reparameterize by $\vec{C}$, the BRDF becomes a function of only 1 variable (θ_i with $\cos\theta_i = \vec{C} \cdot \vec{L}$) instead of 3. Further, the reflected light field can be represented simply by a 2D reflection map $B(\alpha, \beta)$ parameterized by $\vec{C} = (\alpha, \beta)$. Note that we will often use

$\vec{R}$, the reflection of the viewing direction about the surface normal, as a synonym for $\vec{C}$ since that is the most common case; however our analysis applies generally.

For general BRDFs, the radial symmetry property does not hold precisely, so they cannot be reduced exactly to 1D functions, nor can B be written simply as a 2D reflection map. Nevertheless, a reparameterization by the reflection vector still yields compact forms. As can be seen for the $\vec{N} \cdot \vec{H}$ model shown in the lower part of figure 2.4, most of the variation in the BRDF is still over only a single variable (θ_i) after reparameterization, while there is very little variation over θ_o (or ϕ) for fixed θ_i. Further, most of the variation in B remains over (α, β), with only low-frequency variation over the other two variables (θ_o, ϕ_o).

To reparameterize, we simply recenter the BRDF (and the reflection integral) about the reflection vector, rather than the surface normal, as shown in figure 2.4. The reflection vector now takes the place of the surface normal, i.e. $\vec{R} = (\alpha, \beta)$, and the dependence on the surface normal becomes indirect (just as the dependence on $\vec{R}$ is indirect in the standard parameterization). The angles θ'_i and θ'_o are now given with respect to $\vec{R}$ by $\cos\theta'_i = \vec{R} \cdot \vec{L}$ and $\cos\theta'_o = \vec{R} \cdot \vec{V}$, with $B(\alpha, \beta, \theta'_o, \phi'_o)$ a function of $\vec{R} = (\alpha, \beta)$ and (θ'_o, ϕ'_o).

Note that, although this paper does not address general 4D anisotropic BRDFs, reparameterization by the tangent vector $\vec{T}$ can be used in special cases to reduce anisotropic BRDFs to 1D or 2D functions, amenable to treatment by our algorithm pipeline. For instance, consider the Kajiya-Kay [35] model. This BRDF is an anisotropic extension of the Phong model, and depends on $\vec{T} \cdot \vec{L}$ and $\vec{T} \cdot \vec{V}$. We may reparameterize by the tangent vector, just as we did above with the reflection vector. Then, $\cos\theta'_i = \vec{T} \cdot \vec{L}$ and $\cos\theta'_o = \vec{T} \cdot \vec{V}$, with the BRDF being given by $\hat{\rho} = \cos^s(\theta'_i - \theta'_o)$. The BRDF is now a simple 2D function, which is only slightly more complex than the reparameterized 1D Phong BRDF.

An important requirement of our reparameterization is that it be suitable for *both* the BRDF $\hat{\rho}$ and the reflected light field B. Thus, Rusinkiewicz's [74] reparameterization of BRDFs, based on the half angle vector $\vec{H}$, cannot be used since it is unsuitable for the reflected light field. $\vec{H}$ depends on the incident light direction, while B depends only on the the viewing direction and surface normal (or reflection vector). Our approach is motivated by the "reflective transformation" used by Wood et al. [88] to reparameterize the outgoing radiance of surface light fields by $\vec{R}$. However, our final representations differ significantly. In their case, the surface light field is parameterized by object geometry and reflection vector $\vec{R}$, while in our case, the "orientation light field" is parameterized by the reflection vector $\vec{R}$ and viewing vector $\vec{V}$.

5.3.2 4D function representations

Our goal is to compactly represent $B(\alpha, \beta, \theta'_o, \phi'_o)$ in a manner suitable for interactive rendering, while exploiting its characteristics. As noted previously [8], the variation over (α, β) may be rapid (high-frequency), while that over (θ'_o, ϕ'_o) is usually slow (low-frequency), reducing to constant (no variation) for radially symmetric 1D BRDFs. There have been a number of representations for 4D functions proposed in the graphics community, primarily for image-based rendering. The main categories are listed below. Table 5.2 compares tabular, compressed, factored, coefficient-based, as well as our SHRM representation, in terms of simplicity, compactness of the representation, efficiency for rendering, ease of error analysis, and speed for computation (prefiltering).

Explicit tabular representation: We may simply tabulate $B(\alpha, \beta, \theta'_o, \phi'_o)$ on a 4D grid. Cabral et al. [8] use a sparse 2D set of standard 2D reflection maps. However, a very large amount of data will be required to accurately tabulate a 4D light field. Cabral et al. [8] use only 12 reflection maps, trading accuracy for compactness. Kautz and McCool [38] approximate the BRDF as a 2D function $\hat{\rho} = u(\theta'_i, \theta'_o)$ having no azimuthal dependence to create a 3D texture $B(\alpha, \beta, \theta'_o)$. This lower-dimensional representation is more compact, but loses some generality, and can still require a large amount of data. Interactive rendering with these methods usually simply involves a texture lookup and interpolation.

Compressed forms: Compression based on vector-quantization or MPEG-like methods can be used to reduce the size of a tabular representation, as done for surface light fields [60, 88]. or multidimensional animations [27]. Care must be taken that the compressed form can be rendered interactively. Note that both computation of and error analysis on the compressed form require us to first compute the dense uncompressed 4D light field, which can be a computation and data-intensive operation.

Method	Simple	Compact	Rendering	Analysis	Speed
Tabular	Yes	No	Yes	No	No
Compressed	No	Yes	Maybe	*	No
Factored	Yes	Yes	Yes	*	No
Coefficients	Yes	Yes	No	Yes	Yes
SHRM	Yes	Yes	Yes	Yes	Yes

Table 5.2: *Comparison of different 4D representations. The columns stand for simplicity/intuitiveness, compactness, efficiency for rendering, ease of error analysis, and speed of computation (prefiltering). We use $*$ for error analysis of compressed and factored representations because, while error analysis is straightforward, it requires knowledge of a densely sampled 4D light field, and cannot be applied directly.*

Factored representation: Factorization can be seen as a simple compression technique that yields compact results suitable for interactive rendering using texture mapping. Previous methods that can be seen as factorizations include eigen-textures [64], polynomial texture maps [53], and BRDF factorizations [37, 57]. The 4D light field is written as the sum of a few terms, each being the product of two 2D functions (textures),

$$B(\alpha, \beta, \theta_o', \phi_o') = \sum_a g_a(\alpha, \beta) h_a(\theta_o', \phi_o'). \tag{5.4}$$

The trick is to find the factors (2D textures) g_a and h_a so only a small number of terms are needed. Rendering can be done interactively by multiplying together texture maps g and h.

Basis function coefficients: Factored representations can be viewed as a basis-function expansion. To see this, we first introduce a representation purely in terms of basis function coefficients:

$$B(\alpha, \beta, \theta_o', \phi_o') = \sum_a \sum_b c_{ab} d_b(\alpha, \beta) h_a(\theta_o', \phi_o'). \tag{5.5}$$

The basis functions are d_b and h_a with coefficients c_{ab}. We need only store c_{ab}, and can evaluate the basis functions procedurally. This is a simple compact form. However, interactive rendering is difficult since there will usually be a large number of coefficients. We may reduce the number of terms by doing the summation over b to get a factored representation identical to equation 5.4, defining

$$g_a(\alpha, \beta) = \sum_b c_{ab} d_b(\alpha, \beta). \tag{5.6}$$

5.4 Spherical Harmonic Reflection Maps

In this section, we introduce the *spherical harmonic reflection map* or SHRM representation, which is a compact factored representation derived from a spherical harmonic basis function expansion. Figure 5.2 shows an overview of our entire pipeline. S and T stand

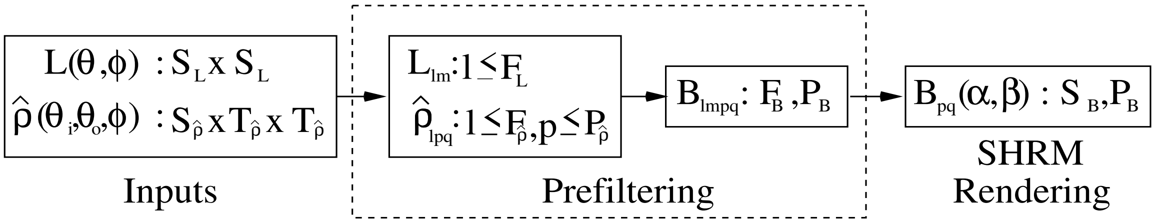

Figure 5.2: *An overview of our entire pipeline. S and T are angular resolutions, while F and P are orders of the spherical harmonic expansions.*

for angular resolutions while F and P stand for orders of the spherical harmonic expansions, which are determined using the theoretical analysis in section 5.5. The inputs to the algorithm are tabulated values of the lighting $L(\theta, \phi)$ and 3D isotropic BRDF $\hat{\rho}(\theta_i', \theta_o', \phi)$.

We then use our fast prefiltering algorithm, described in detail in section 5.6, to compute the SHRM. This is done by first computing the spherical harmonic lighting coefficients L_{lm} and BRDF coefficients $\hat{\rho}_{lpq}$. We then use the spherical frequency-space convolution formula derived in chapter 2, which may be viewed as the frequency domain analog of equation 5.2, to compute reflected light field coefficients B_{lmpq} (c.f. equation 2.55),

$$B_{lmpq} = \Lambda_l L_{lm} \hat{\rho}_{lpq}. \tag{5.7}$$

Now, if we want to expand the coefficients to get the entire reflected light field, we know that (c.f. equation 2.54)

$$\begin{aligned}
B(\alpha,\beta,\theta_o',\phi_o') &= \sum_{l,m,p,q} \Lambda_l L_{lm}\hat{\rho}_{lpq}\Lambda_l^{-1} D^l_{mq}(\alpha,\beta) Y_{pq}(\theta_o',\phi_o') \\
&= \sum_{lmpq} L_{lm}\hat{\rho}_{lpq}\left(D^l_{mq}(\alpha,\beta)Y_{pq}(\theta_o',\phi_o')\right). \qquad (5.8)
\end{aligned}$$

We expand the right-hand side partially (over indices l and m), defining

$$B_{pq}(\alpha,\beta) = \sum_{l,m} L_{lm}\hat{\rho}_{lpq} D^l_{mq}(\alpha,\beta). \qquad (5.9)$$

Finally, we may write down the equation for the SHRM as

$$B(\alpha,\beta,\theta_o,\phi_o) = \sum_{p=0}^{P_B}\sum_{q=-p}^{p} B_{pq}(\alpha,\beta)Y_{pq}(\theta_o',\phi_o'). \qquad (5.10)$$

In this equation, $B_{pq}(\alpha,\beta)$ is one coefficient in the SHRM, and $P_B \geq 0$ is the maximum order of the expansion, with the SHRM containing a total of $(P_B+1)^2$ terms. Figure 5.3 illustrates the idea behind SHRMs. Each pixel (α,β) in a reflection (cube)map has a particular distribution of outgoing radiance $B(\alpha,\beta,\theta_o',\phi_o')$. This distribution is encoded by the SHRM as a spherical harmonic expansion in (θ_o',ϕ_o'), with coefficients $B_{pq}(\alpha,\beta)$. For the special case of radially symmetric 1D BRDFs, there is no dependence on (θ_o',ϕ_o') after reparameterization, so we need only the DC or constant term $B_{00}(\alpha,\beta)$, and the SHRM reduces to a simple 2D reflection map.

So far, we have considered *local* SHRMs, depending on local outgoing angles (θ_o',ϕ_o'), which are different for each (α,β). It is often convenient to assume the viewer is distant and compute *global* SHRMs, dependent on a global viewing direction (θ_o,ϕ_o).

$$\hat{B}(\alpha,\beta,\theta_o,\phi_o) = \sum_{p=0}^{P_B}\sum_{q=-p}^{p} \hat{B}_{pq}(\alpha,\beta)Y_{pq}(\theta_o,\phi_o). \qquad (5.11)$$

The advantage of equation 5.11 over equation 5.10 lies in ease of evaluation for rendering, since $Y_{pq}(\theta_o,\phi_o)$ can be evaluated once per frame, instead of per pixel. In fact, we will show in section 5.7.5 that this allows global SHRMs to be rendered using a single dynamic

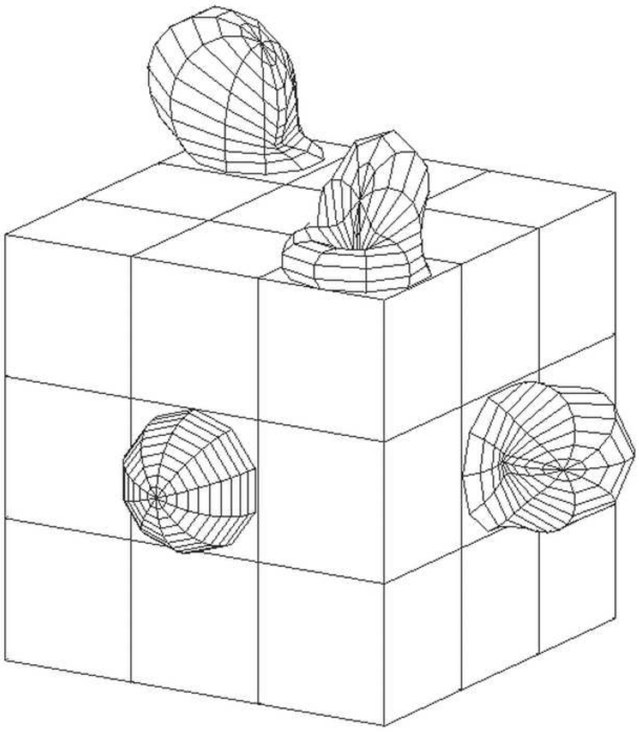

Figure 5.3: *The idea behind SHRMs. Each pixel (α, β) in the reflection cubemap has some distribution of reflected light. This distribution is encoded as a low-order spherical harmonic expansion in (θ'_o, ϕ'_o) for every pixel, with coefficients $B_{pq}(\alpha, \beta)$. To avoid clutter, this diagram uses only a 3x3 resolution in the reflection map and shows the outgoing distribution for only four of the pixels.*

reflection map, with standard reflection mapping hardware.

We still need to know how to determine global SHRM coefficients $\hat{B}_{pq}(\alpha, \beta)$. The spherical convolution formula in equations 5.7 or 2.55 apply only to local SHRMs. However, we may rotate coefficients to compute the global SHRM. We make use of equations 5.3 and 5.10, with the subscript q changed to s for later convenience,

$$\hat{B}\left(\alpha, \beta, \theta_o, \phi_o\right) = \sum_{p=0}^{P_B} \sum_{s=-p}^{p} B_{ps}(\alpha, \beta) Y_{ps}\left(R_{\alpha,\beta}^{-1}(\theta_o, \phi_o)\right). \quad (5.12)$$

From this expression, we can determine the modified (rotated) coefficients of the global SHRM separately for each orientation (α, β). For this, we need to know how to apply the inverse rotation in the equation above, analogous to the forward rotation formula used in equation 2.29. We may derive

$$\begin{aligned} Y_{ps}\left(R^{-1}_{\alpha,\beta}(\theta_o,\phi_o)\right) &= Y_{ps}\left(R_y(-\alpha)R_z(-\beta)\{\theta_o,\phi_o\}\right) \\ &= \sum_{q=-p}^{p} D^p_{sq}(-\alpha)Y_{pq}(R_z(-\beta)\{\theta_o,\phi_o\}) \\ &= \sum_{q=-p}^{p} D^p_{sq}(-\alpha)e^{-Iq\beta}Y_{pq}(\theta_o,\phi_o). \end{aligned} \quad (5.13)$$

Finally, putting equation 5.12 and 5.13, we obtain

$$\hat{B}_{pq}(\alpha,\beta) = \sum_{s=-p}^{p}\left(D^p_{sq}(-\alpha)e^{-Iq\beta}\right)B_{ps}(\alpha,\beta). \quad (5.14)$$

Advantages: SHRMs are a hybrid form, midway between a pure coefficient-based approach, and an explicit tabular representation. We believe this is a good point in the design space, and our representation has the following significant advantages:

- **Compact, Efficient and Accurate:** An important observation from the theoretical analysis is that for essentially all BRDFs, a very low value of P_B (usually ≤ 3) suffices for high accuracy. This is the formal basis for using a low order spherical harmonic expansion in the SHRM, and ensures that our representation is very compact and accurate compared to previous approaches, as well as being efficient to evaluate for real-time rendering.

- **Error analysis and number of coefficients/resolutions:** Unlike for other compression and factorization techniques, the error analysis in section 5.5 does not first require computation of a dense 4D reflected light field, and allows us to easily determine the correct order P_B of the spherical harmonic expansion and the resolutions of the reflection maps.

- **Rapid computation:** In section 5.6, we show how the SHRM can be computed using frequency domain prefiltering, orders of magnitude faster than previous approaches.

5.5 Analysis of sampling rates/resolutions

In this section, we present our framework for analyzing the required sampling rates, i.e. the number of coefficients needed in our spherical harmonic expansions. At the end of the section, we will justify the SHRM representation based on our analysis.

The sampling rates will depend on the frequency content of the lighting and BRDF. Figure 5.4 shows spheres rendered with progressively blurred illumination (along the y axis) and a progressively more diffuse BRDF (along the x axis). It can be seen that the highest frequencies in the reflected light field are determined approximately by the minimum of the highest frequencies in the lighting and BRDF. This is not surprising, since we may view the BRDF as a low pass filter acting on the lighting signal.

As summarized in figure 5.2, we assume the input lighting $L(\theta, \phi)$ is represented on an $S_L \times S_L$ grid, where S_L is the grid angular resolution, and that the 3D isotropic BRDF $\hat{\rho}(\theta_i', \theta_o', \phi)$ is represented on a grid of size $S_{\hat{\rho}} \times T_{\hat{\rho}} \times T_{\hat{\rho}}$ where $S_{\hat{\rho}}$ is the angular resolution with respect to θ_i' and $T_{\hat{\rho}}$ is the angular resolution with respect to (θ_o, ϕ). For simplicity, we will consider the lighting and BRDF to be represented in latitude-longitude form, i.e. simply as tabulated values on an equally-spaced grid $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. Converting to less distorted forms like cubemaps for practical use on graphics hardware is a straightforward table lookup.

Our prefiltering algorithm computes the lighting coefficients L_{lm} to order F_L (i.e. F_L is the maximum value of l) and BRDF coefficients $\hat{\rho}_{lpq}$ to orders $F_{\hat{\rho}}$ and $P_{\hat{\rho}}$ (i.e. $l \leq F_{\hat{\rho}}$ and $p \leq P_{\hat{\rho}}$). The light field coefficients B_{lmpq} are computed to orders F_B and P_B. Finally, we generate the SHRM $B_{pq}(\alpha, \beta)$, with the angular size in (α, β) being $S_B \times S_B$, and the spherical harmonic expansion up to order P_B. Radially symmetric 1D BRDFs can be seen as special cases of this general framework with $T_{\hat{\rho}} = 1$, and $P_{\hat{\rho}} = P_B = 0$.

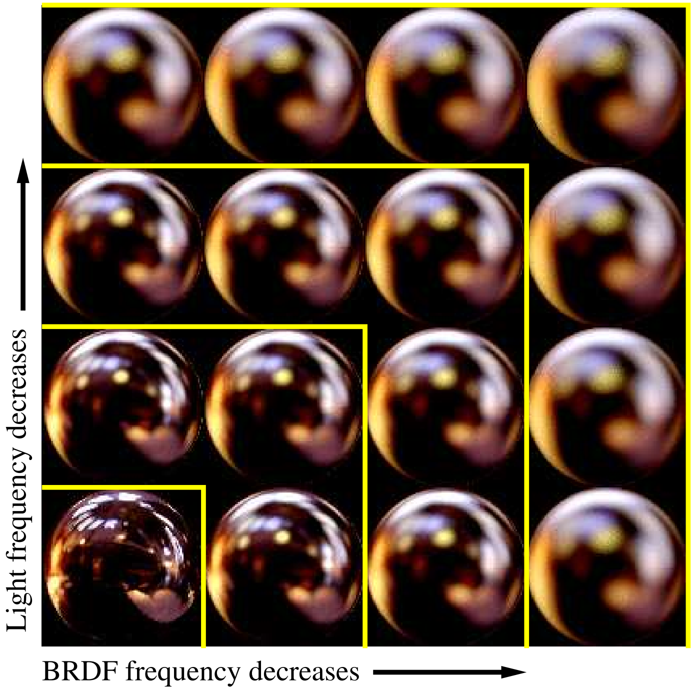

Figure 5.4: *Renderings with different lighting and BRDF conditions. The highest frequency in the images is approximately the minimum of the highest frequencies in the lighting and BRDF. Specifically, all spheres inside a yellow delimiter look similar.*

5.5.1 Order of spherical harmonic expansions

We now analyze required orders F and P in our spherical harmonic expansions. First, consider the lighting. The total energy of the lighting signal is given by

$$\int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} L^2(\theta, \phi) \sin\theta \, d\theta d\phi = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mid L_{lm} \mid^2 . \tag{5.15}$$

Once we compute the total energy in the lighting, we can estimate the error ϵ in an order F_L spherical harmonic expansion by considering what fraction of the total lighting energy is captured up to order F_L. To obtain an accuracy $1-\epsilon$, we require that

$$\sum_{l=0}^{F_L} \sum_{m=-l}^{l} \mid L_{lm} \mid^2 \geq (1-\epsilon) \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} L^2(\theta, \phi) \sin\theta \, d\theta d\phi. \tag{5.16}$$

For any given input illumination $L(\theta, \phi)$ and frequency F_L, it is easy to determine what the error ϵ is, and check if it is below threshold. Conversely, if we fix the error threshold ϵ, we can compute the required sampling rate or order F_L as the minimum frequency for which the above equation holds. Note that the number of coefficients required measures the frequency width of the signal, and ϵ measures the missing (residual) information content.

A similar method may be used for analyzing the BRDF. We give below the separate results for the 1D radially symmetric and 3D isotropic cases,

$$\sum_{l=0}^{F_{\hat{\rho}}} \mid \hat{\rho}_l \mid^2 \geq (1-\epsilon) 2\pi \int_{\theta_i=0}^{\pi/2} \hat{\rho}^2(\theta_i) \sin\theta_i \, d\theta_i. \tag{5.17}$$

$$\begin{aligned} &\sum_{l=0}^{F_{\hat{\rho}}} \sum_{p=0}^{P_{\hat{\rho}}} \sum_{q=-\min(l,p)}^{\min(l,p)} \mid \hat{\rho}_{lpq} \mid^2 \geq \\ &(1-\epsilon) 2\pi \int_0^{\pi} \int_0^{\pi} \int_0^{2\pi} \hat{\rho}^2(\theta_i, \theta_o, \phi) \sin\theta_i \sin\theta_o \, d\theta_i d\theta_o d\phi. \end{aligned} \tag{5.18}$$

The remaining issue is how to combine the information for lighting and BRDFs to determine appropriate orders for the reflected light field B. We list below two possible approaches.

- **Minimum of orders or errors:** Consider the case where $\epsilon = 0$ for either the lighting or BRDF, i.e. one or both is bandlimited. The reflected light field is then exactly reproduced by using an expansion to order $(F_B, P_B) = (\min(F_L, F_{\hat{\rho}}), P_{\hat{\rho}})$. This formalizes the intuition that we need to sample densely enough to catch the highest frequency present simultaneously in both the lighting signal and BRDF filter. This analysis does not apply rigorously when neither signal is bandlimited, but simply decays with increasing frequency. Nevertheless, in practice, taking the minimum of orders for a given error ϵ is still a good heuristic. Conversely, for a given order of expansion, we can estimate the error $\epsilon_B = \min(\epsilon_L, \epsilon_{\hat{\rho}})$.

  Since the lighting signal usually contains substantial high frequency content, while the BRDF acts as a low-pass filter, this method often reduces simply to capturing $1-\epsilon$ of the BRDF energy, i.e. choosing $F_B, P_B = F_{\hat{\rho}}, P_{\hat{\rho}}$ or setting $\epsilon_B = \epsilon_{\hat{\rho}}$.

- **Bound residual energy:** For completeness, we discuss a more rigorous numerical scheme, which can be proven to give conservative estimates. The scheme is based on bounding the residual unaccounted for energy in the reflected light field. One disadvantage of this method is that, unlike the previous method, we first need to actually calculate the coefficients B_{lmpq} of the reflected light field. Thus, this method is most useful as a final sanity check on the validity of the earlier heuristic.

To derive this bound, we consider an expansion to order F with errors for lighting and BRDF given by ϵ_L and $\epsilon_{\hat{\rho}}$. Denote the total BRDF and lighting energies by $\hat{\rho}_{tot}$ and L_{tot}. Since the reflected light field coefficients are simply a product of lighting and BRDF terms, the worst case for the residual energy occurs when it is all concentrated in mode $F+1$. This residual energy, denoted by B_{res}, and a conservative error estimate ϵ_B are

$$\begin{aligned} B_{res} &= \Lambda_{F+1}\left(\epsilon_L L_{tot}\right)\left(\epsilon_{\hat{\rho}}\hat{\rho}_{tot}\right) \\ \epsilon_B &\leq \frac{B_{res}}{\sum_{l=0}^{F}\sum_{m=-l}^{l}\left|B_{lm}\right|^2 + B_{res}}. \end{aligned} \tag{5.19}$$

This is for the radially symmetric case; in general, we simply use B_{lmpq} in place of B_{lm}. Note that ϵ_B tends to 0 as B_{res} tends to 0. But the latter quantity is a product of $\epsilon_{\hat{\rho}}$ and ϵ_L, and therefore always tends to 0 as F increases.

We use a simple example to illustrate these methods. For a particular illumination (the Grace Cathedral), and a Phong BRDF (exponent $s = 32$), we computed approximations to the lighting, BRDF, and reflected light field for increasing values of order $F = F_L = F_{\hat{\rho}} = F_B$. Since the BRDF is radially symmetric, $P = P_{\hat{\rho}} = P_B = 0$. We also computed the reflected light field accurately, by using a very high order $F = 30$, so we could determine the errors of lower-order approximations. Figure 5.5 plots the accuracy (top curve) of an order F approximation of B, as well as estimates of this accuracy obtained by taking the minimum of BRDF and light errors at order F, and by bounding the residual energy. We see that both accuracy estimates are conservative but fairly tight, especially for small errors or high accuracies (at higher frequencies). Further, taking the minimum of lighting and BRDF errors is almost always equivalent simply to using the BRDF error. Therefore, we choose the simplest approach of using the BRDF error, requiring $\epsilon_{\hat{\rho}}$ be lower than a tolerance that the user has selected.

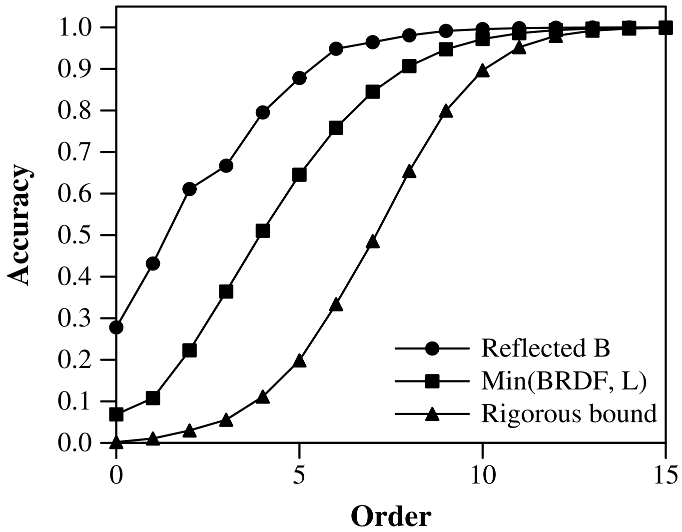

Figure 5.5: *Accuracy $(1-\epsilon)$ versus frequency F for an order F approximation of the reflected light field B , and estimates of that accuracy obtained by taking the minimum error for BRDF and lighting, and by using the conservative bound based on residual energy. We have not separately plotted using the BRDF error only, as this gives almost exactly the same curve as taking the minimum error for BRDF and lighting.*

5.5.2 Justification for SHRM representation

We seek to determine the best point in the spectrum of time/space or angular/frequency tradeoffs. For this, we must understand how to relate angular space resolutions S and T to frequency-space orders F and P. As a simple illustration, consider irradiance maps from Lambertian BRDFs. It has been shown [2, 72] that an order 2 spherical harmonic expansion suffices. However, a 3×3 irradiance map will clearly be inadequate. In practice, irradiance maps are usually represented at angular resolutions higher than 16×16. Experimenting with different resolutions, we have found that in general, one requires $S \sim 10F$ (and $T \sim 10P$).

Therefore, a significantly more compact size for B is obtained using spherical harmonic coefficients rather than an explicit 4D tabular representation. The other extreme in the spectrum of time-space tradeoffs—using a purely coefficient-based approach—is also usually[1] undesirable. Efficient rendering of 2D reflection maps having high frequency content, such as specular reflection maps from Phong BRDFs, is difficult directly from the spherical harmonic expansion, since $O(F^2)$ terms must be added per pixel, with F generally larger than 10. Rendering the 4D light field purely from coefficients is even harder, requiring $O(F^2P^2)$ terms.

Hence, we believe an intermediate representation, allowing for both compact representation, and fast rendering, is optimal. In order to determine the best representation for B, we must know the common values for orders F and P (and hence resolutions S and T). Our results in section 5.7 show that for practically all currently available analytic and measured BRDFs, values of $F \leq 30$ and $P \leq 5$ suffice for an accuracy greater than 90%. Therefore, it is best to encode the view dependence (θ_o', ϕ_o') as a compact (and easily evaluated) spherical harmonic expansion consisting of $(P+1)^2$ terms, while explicitly representing the high-frequency dependence on (α, β). This is the approach taken by SHRMs, where each pixel (α, β) stores coefficients $B_{pq}(\alpha, \beta)$ of an order P spherical harmonic expansion.

5.6 Prefiltering

We now describe our efficient frequency space prefiltering algorithms to create the SHRM and efficiently implement the pipeline in figure 5.2. Frequency space representations allow for very efficient computations and are also very compact, requiring evaluation of fewer terms. Hence our algorithms are much faster than previous angular-space methods. We will present an analysis of the computational complexity of our algorithms, and end this section by validating our conclusions on the Phong BRDF.

[1]For very diffuse BRDFs ($F_{\hat{\rho}}$ and $P_{\hat{\rho}}$ both very small), a purely coefficient-based approach may be acceptable. The most notable example is the Lambertian BRDF ($F_{\hat{\rho}} = 2$, $P_{\hat{\rho}} = 0$), where a 9 term spherical harmonic expansion suffices, as seen in the previous chapter.

5.6.1 Main steps and insights

Our prefiltering method has two main components. First, we must efficiently convert between angular and frequency space descriptions. Second, we must efficiently compute coefficients of the reflected light field from lighting and BRDF coefficients. Both components can be performed rapidly because of the insights below. The algorithm itself is just a direct three step efficient implementation of the pipeline of figure 5.2. Implementation details, and the time complexities of the various steps, are found in section 5.6.2.

Linear time complexity of convolution formula: The reflected light field coefficients B_{lmpq} can be computed in time *linear* in the number of output coefficients B_{lmpq} simply by applying the spherical convolution formula in equation 5.7.

Fast conversion to and from spherical harmonics: We still need to convert from an angular space representation of $L(\theta,\phi)$ and $\hat{\rho}(\theta_i,\theta_o,\phi)$ to the spherical harmonic coefficients, as well as generate the SHRM from B_{lmpq}. As an example, consider computation of lighting coefficients L_{lm}. For any l, m we have

$$L_{lm} = \int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} L(\theta,\phi)Y_{lm}^{*}(\theta,\phi)\sin\theta\, d\theta d\phi. \tag{5.20}$$

The cost of performing this integral is $O(S_L^2)$. Since we must do this for all coefficients, it would appear the total cost would be $O(F_L^2 S_L^2)$. In fact, we can amortize the cost to compute all the coefficients in $O(F_L S_L^2)$ time by writing the spherical harmonics as products of functions in θ and ϕ, and then separating the computations in θ and ϕ. The basic idea is to compute in succession:

$$\begin{aligned}
\forall m,\theta: \quad L_m(\theta) &= \int_{\phi=0}^{2\pi} L(\theta,\phi)e^{-Im\phi}\, d\phi \\
\forall l,m: \quad L_{lm} &= \int_{\theta=0}^{\pi} L_m(\theta) f_{lm}(\theta)\sin\theta\, d\theta.
\end{aligned} \tag{5.21}$$

Here, the spherical harmonic $Y_{lm}(\theta,\phi) = f_{lm}(\theta)e^{Im\phi}$. The first step involves a loop over $(2F_L{+}1)S_L$ elements, each step of which requires numerical integration by adding together S_L values. Thus, the cost is $O(F_L S_L^2)$. A similar argument shows that the second step takes time of $O(F_L^2 S_L)$. Since $F_L < S_L$, the first step dominates, and the total complexity is $O(F_L S_L^2)$. Note that a similar approach can also be used to efficiently compute BRDF coefficients, and for the inverse transform to compute the SHRM. Fast spherical harmonic transform methods [61], analogous to the Fast Fourier Transform, may reduce the cost further[2] to logarithmic in F_L, i.e $O(S_L^2 \log^2 S_L) \sim O(S_L^2 \log^2 F_L)$. However, these methods are complicated, and although asymptotically faster, have relatively large constant cost factors. Therefore, they are unlikely to be significantly faster for the low frequencies $F \sim 30$ relevant for us.

5.6.2 Prefiltering Algorithm

Below, we give full details of our prefiltering algorithm, which is a three-step process that implements the pipeline in figure 5.2.

Step 1. Compute lighting and BRDF coefficients: We go from the inputs $L(\theta, \phi)$ and $\hat{\rho}(\theta_i, \theta_o, \phi)$ to the spherical harmonic coefficients L_{lm} and $\hat{\rho}_{lpq}$. We first compute the spherical harmonic coefficients of the BRDF using the following three step algorithm.

$$\begin{aligned}
\hat{\rho}_q(\theta_i, \theta_o) &= 2\pi \int_0^{2\pi} \hat{\rho}(\theta_i, \theta_o, \phi) \cos q\phi \, d\phi \\
\hat{\rho}_{pq}(\theta_i) &= \int_0^{\pi} \hat{\rho}_q(\theta_i, \theta_o) f_{pq}(\theta_o) \sin\theta_o \, d\theta_o \\
\hat{\rho}_{lpq} &= \int_0^{\pi} \hat{\rho}_{pq}(\theta_i) f_{lq}(\theta_i) \sin\theta_i \, d\theta_i.
\end{aligned} \tag{5.22}$$

The computational costs of the three terms in the above sequence are given by, respectively, $O(P_{\hat{\rho}} S_{\hat{\rho}} T_{\hat{\rho}}^2)$, $O(P_{\hat{\rho}}^2 S_{\hat{\rho}} T_{\hat{\rho}})$, and $O(F_{\hat{\rho}} P_{\hat{\rho}}^2 S_{\hat{\rho}})$. Since $P_{\hat{\rho}} < T_{\hat{\rho}}$, the first term dominates the second, and the net cost is $O\left(P_{\hat{\rho}} S_{\hat{\rho}} (T_{\hat{\rho}}^2 + F_{\hat{\rho}} P_{\hat{\rho}})\right)$. For most non radially-symmetric BRDFs, $T_{\hat{\rho}}^2 > F_{\hat{\rho}} P_{\hat{\rho}}$ (for instance, use $T_{\hat{\rho}} \sim 10$, $F_{\hat{\rho}} \sim 20$ and $P_{\hat{\rho}} \sim 3$), so the first term dominates and the total cost is $O(P_{\hat{\rho}} S_{\hat{\rho}} T_{\hat{\rho}}^2)$.

If our error tolerance ϵ is satisfied, we see how far $P_{\hat{\rho}}$ can be reduced to still satisfy the error tolerance, and then also reduce $F_{\hat{\rho}}$ as much as possible. We can then set F_B and

[2] Since $S_L \sim 10F_L$, $\log S_L \sim \log F_L$. Also note that simply using an FFT in step 1 will not suffice, since step 2 does not have logarithmic complexity.

P_B according to the minimal values of $F_{\hat{\rho}}$ and $P_{\hat{\rho}}$. If the error is initially larger than ϵ, we repeat the algorithm with larger values for $F_{\hat{\rho}}$ and $P_{\hat{\rho}}$. Since computing BRDF coefficients is not the dominant algorithm cost, this recomputation does not significantly affect the total time, nor does using large initial values for $F_{\hat{\rho}}$ and $P_{\hat{\rho}}$.

Finally, we compute the lighting coefficients in time $O(F_B S_L^2)$. Note that we have already determined F_B, so we are not required to consider higher frequencies for the lighting, which is why we use F_B instead of F_L.

Step 2. Find reflected light field coefficients: We now find $B_{lmpq} = L_{lm}\hat{\rho}_{lpq}$ in time $O(F_B^2 P_B^2)$. Note that we have dropped the constant Λ_l compared to equation 5.7 to avoid dealing with the factors of Λ_l^{-1} and Λ_l.

Step 3. Compute SHRM: We now compute the local SHRM. We need to compute

$$B_{pq}(\alpha, \beta) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} B_{lmpq} \left(D_{mq}^{l}(\alpha) e^{Im\beta} \right), \tag{5.23}$$

which can be done efficiently as shown below,

$$\begin{aligned} B_{mpq}(\alpha) &= \sum_{l=|m|}^{F_B} B_{lmpq} D_{mq}^{l}(\alpha) \\ B_{pq}(\alpha, \beta) &= \sum_{m=-F_B}^{F_B} B_{mpq}(\alpha) e^{Im\beta}. \end{aligned} \tag{5.24}$$

From this, we can compute the *global* SHRM $\tilde{B}_{pq}(\alpha, \beta)$ using equation 5.14. The costs of the two terms in the above sequence are $O(F_B^2 P_B^2 S_B)$ and $O(F_B P_B^2 S_B^2)$. Since $S_B > F_B$, the net cost is $O(F_B P_B^2 S_B^2)$. The cost for this step is also the dominant cost for the entire algorithm.

Radially symmetric BRDFs: For the special case of BRDFs like Lambertian and Phong models, $T_{\hat{\rho}} = 1$ and $P_{\hat{\rho}} = P_B = 0$. Technically, the complexity formulae above should use $P + 1$ instead of P, to yield meaningful results for radially symmetric BRDFs. For these models, step 1 takes time of $O(F_{\hat{\rho}} S_{\hat{\rho}})$ to compute BRDF coefficients $\hat{\rho}_l$, and time $O(F_B S_L^2)$

to compute lighting coefficients L_{lm}. Step 2 takes $O(F_B^2)$ time. Finally, the SHRM in step 3 includes only the constant term and is therefore a simple reflection map $B(\alpha, \beta)$, computed in time $O(F_B S_B^2)$. The dominant cost here is to convert to and from spherical harmonic representations. Assuming we downsample the environment map if necessary so $S_L \sim S_B$, the total time is $O(F_B S_B^2)$ or $O(F_B)$ per output image pixel.

Conversion between SHRMs and explicit forms: It is possible to incorporate the prefiltering and rendering phases of our algorithm separately into existing systems. SHRMs may be created from explicit representations simply by fitting coefficients or integrating. If the implementer wants to use only our fast prefiltering method, but render using previous explicit representations, they can compute tabular representations from SHRMs. Cabral's twelve prerendered reflection maps may be computed very rapidly using equation 5.35, with $(\tilde{\theta}_o, \tilde{\phi}_o)$ set to vertices of an icosahedron. Kautz and McCool's [38] 3D texture is computed by expanding

$$B(\alpha, \beta, \theta_o) = \sum_{p=0}^{P_B} B_{p0}(\alpha, \beta) Y_{p0}(\theta_o). \tag{5.25}$$

This takes time $O(P_B)$ per output texel. Using fast conversion methods, we can also explicitly generate $T_B \times T_B$ reflection maps (a full 4D light field) in time $O(P_B)$ per output pixel, for a total cost of $O(P_B T_B^2 S_B^2)$.

5.6.3 Computational complexity

The cost of previous angular domain algorithms is $O(W S_L^2)$ per pixel in the output, since they perform a hemispherical integral for each pixel. Here, W is the fraction of the illumination pixels that need be considered, corresponding to the angular width of the BRDF, with $W \to 0$ for a mirror, and $W = 1/2$ if one considers the entire visible hemisphere. In appendix C, we derive the cost for our frequency space algorithm[3], which is much lower, being $O(F_B)$ or $O(P_B)$ per pixel. Table 5.3 summarizes our main results.

[3] If we were to use fast spherical harmonic transform methods [61], the asymptotic complexity per output pixel would be $O(\log^2 F_B)$ or $O(\log^2 P_B)$ instead.

Type	Angular		Frequency	
	Cost	/pixel	Cost	/pixel
1D BRDF	$WS_L^2S_B^2$	WS_L^2	$F_BS_B^2$	F_B
SHRM			$F_BP_B^2S_B^2$	F_B
3D BRDF	$WS_L^2T_B^2S_B^2$	WS_L^2	$P_BT_B^2S_B^2$	P_B

Table 5.3: *Computational complexity of prefiltering. We show both total costs and costs per pixel. The angular costs correspond to hemispherical integration, with W being the BRDF angular width. The frequency space costs are for our method.*

Radially Symmetric 1D BRDFs: The output reflection map size is $S_B \times S_B$. Standard hemispherical integration is *quadratic* in the output size, since we must examine $O(WS_L^2)$ illumination pixels per output pixel and[4] $S_L \geq S_B$. By contrast, our frequency space prefiltering algorithm requires only $O(F_B)$ time per output pixel. Since $F_B \ll S_B < S_L$, this method is only slightly *super-linear*, being substantially *sub-quadratic* in the output size.

SHRM creation: The SHRM is a new representation, not created by traditional prefiltering algorithms. We can still compare our prefiltering cost with the theoretical minimal complexity. As for 1D BRDFs, the cost of our method is $O(F_B)$ per output pixel per coefficient.

3D BRDFs, explicit reflection maps: In order to compare to angular domain methods, we must produce the same output. We can use the SHRM to explicitly compute a set of $T_B \times T_B$ reflection maps (with $T_B \sim 10P_B$), similar to the explicit representations of Cabral et al. [8] or Kautz and McCool [38][5]. The cost of traditional prefiltering remains $O(WS_L^2)$ per output pixel. On the other hand, our method takes $O(P_B)$ time per pixel. Since $P_B \leq 3$ in most cases, it can be regarded a constant. Hence, our method is *quasi-linear* in the output size. This is a speedup of three to four orders of magnitude—the difference between near-interactive computation in a few seconds, and prefiltering times in hours reported by Cabral et al. [8] and other authors.

[4]One way to compute F_B is $\min(F_L, F_{\hat{\rho}})$ so $F_B \leq F_L$ and $S_B \leq S_L$.
[5]Since their representation is 3D, we should compute only T_B reflection maps.

5.6.4 Validation with Phong BRDF

In this subsection, we validate our theoretical computational complexity analysis on the simple radially symmetric Phong model. In this case, $P_{\hat{\rho}} = P_B = 0$ and the SHRM reduces to a standard 2D reflection map $B(\alpha, \beta)$. In the results section, we show timings, including for more general 3D isotropic BRDFs.

The normalized and reparameterized Phong BRDF is defined by

$$\hat{\rho} = \frac{s+1}{2\pi} \cos^s \theta_i, \tag{5.26}$$

where $\cos^s \theta_i = (\vec{R} \cdot \vec{L})^s$. BRDF coefficients $\hat{\rho}_l$ can be derived analytically [73], and an accurate approximation is

$$\hat{\rho}_l \approx \Lambda_l^{-1} \exp\left[-\frac{l^2}{2s}\right]. \tag{5.27}$$

In the frequency domain, the order $F = F_B = F_{\hat{\rho}}$ for an error ϵ is found by the following sequence of steps,

$$\begin{aligned}
\sum_{l=0}^{F} \hat{\rho}_l^2 &= (1-\epsilon) \sum_{l=0}^{\infty} \hat{\rho}_l^2 \\
\int_0^F l e^{-l^2/s}\, dl &\approx (1-\epsilon) \int_0^{\infty} l e^{-l^2/s}\, dl \\
1 - e^{-F^2/s} &\approx 1 - \epsilon \\
F &\approx \sqrt{-s \log \epsilon}.
\end{aligned} \tag{5.28}$$

Ignoring the constant $\sqrt{-\log \epsilon}$, the prefiltering cost of our frequency space algorithm is therefore

$$C_f = O(F_B S_B^2) = O(S_B^2 \sqrt{s}). \tag{5.29}$$

In the angular domain, we may truncate the BRDF, so $(1-\epsilon)$ of the angular domain energy lies in $\theta_i \leq \theta_i^*$. We find the angular width W and θ_i^* by the following steps,

$$\begin{aligned}
\int_0^{\theta_i^*} \cos^s \theta_i \sin \theta_i \, d\theta_i &= (1-\epsilon) \int_0^{\pi/2} \cos^s \theta_i \sin \theta_i \, d\theta_i \\
1 - \cos^{s+1} \theta_i^* &= 1 - \epsilon \\
\cos \theta_i^* &= \epsilon^{1/(s+1)} \\
W = \frac{2\pi}{4\pi} \int_0^{\theta_i^*} \sin \theta_i \, d\theta_i &= \frac{1}{2} \left(1 - \epsilon^{1/(s+1)}\right).
\end{aligned} \tag{5.30}$$

We now assume s is large ($1/s \to 0$) and perform a Taylor series expansion,

$$W \sim \left(1 - \epsilon^{1/(s+1)}\right) = \left(1 - \left[1 + \frac{\log \epsilon}{s+1}\right]\right) \approx \frac{-\log \epsilon}{s}. \tag{5.31}$$

Ignoring the factor of $\log \epsilon$, the angular domain prefiltering cost is

$$C_a = O(W S_L^2 S_B^2) = O\left(\frac{S_L^2 S_B^2}{s}\right). \tag{5.32}$$

We note that the frequency space cost C_f increases with increasing Phong exponent as $\sqrt{s}$, while the angular space cost C_a decreases with increasing Phong exponent as $1/s$. This is entirely expected, since sharp specular surfaces (large s) have a BRDF which is very local in the angular domain but requires a large number of coefficients to represent in the frequency domain. Conversely, rough surfaces (small s) are very easily handled in the frequency domain, but their BRDFs have a large angular width. Therefore, for $s < s^*$, our frequency domain methods are to be preferred and for $s > s^*$, conventional angular domain techniques are preferable. s^* can be found by equating equations 5.29 and 5.32,

$$S_B^2 \sqrt{s^*} \sim S_L^2 S_B^2 / s^* \Rightarrow s^* \sim S_L^{4/3}. \tag{5.33}$$

What does this mean numerically? Assume a small size of $S_L = 100$. We then obtain $s^* \approx 464$. Therefore, in essentially all practical cases of interest, our frequency domain algorithm is superior to conventional angular domain methods, often by one to two orders of magnitude. Of course, the actual numerical value for s^* depends on the constant cost factors associated with the respective implementations. Our empirical tests, discussed in section 5.7.4, show that the practical value of s^* is actually even higher than predicted.

5.7 Results

We have tested our method using a number of different lighting conditions and BRDFs. This section reports our main results.

5.7.1 Number of coefficients for analytic BRDFs

The practical values of the orders in the spherical harmonic expansion of the BRDF $F_{\hat{\rho}}$ and $P_{\hat{\rho}}$ (and hence F_B and P_B) will depend on the form of the BRDF, with slowly varying BRDFs requiring fewer coefficients. As a basic test to determine reasonable empirical values, we considered three general analytic BRDFs.

Microfacet: Consider a simplified microfacet [84] model,

$$\hat{\rho} = \frac{1}{4\pi\sigma^2} e^{-(\theta_h/\sigma)^2}, \tag{5.34}$$

where $\theta_h = \cos^{-1}(\vec{N} \cdot \vec{H})$ is the angle between the normal and the half-angle vector. Approximations to reflection maps with different values of $P = P_{\hat{\rho}} = P_B$ are shown in figure 5.6. As expected, the accuracy improves as we use higher values of P. Specifically, $P = 2$ suffices to produce very accurate results, with the BRDF error $\epsilon < .03$. Recall from section 5.5.1 that we use the BRDF accuracy as a conservative estimate of the accuracy of the reflected light field. For this BRDF, $F = F_{\hat{\rho}} = F_B$ is given approximately by $F \sim \sigma^{-1}$, and ranges from 10 to 30 for common values of $\sigma \sim 0.1$. These values of F and P are typical for most BRDFs. In general, P is very small, while F is usually much larger.

Lafortune BRDF: We also tested the model of [44], with coefficients obtained from the skin measurements of Marschner et al. [55]. Although the behavior is more interesting, with much stronger specularities exhibited toward grazing angles, a value of $P = 4$ still suffices for an error $\epsilon < .03$.

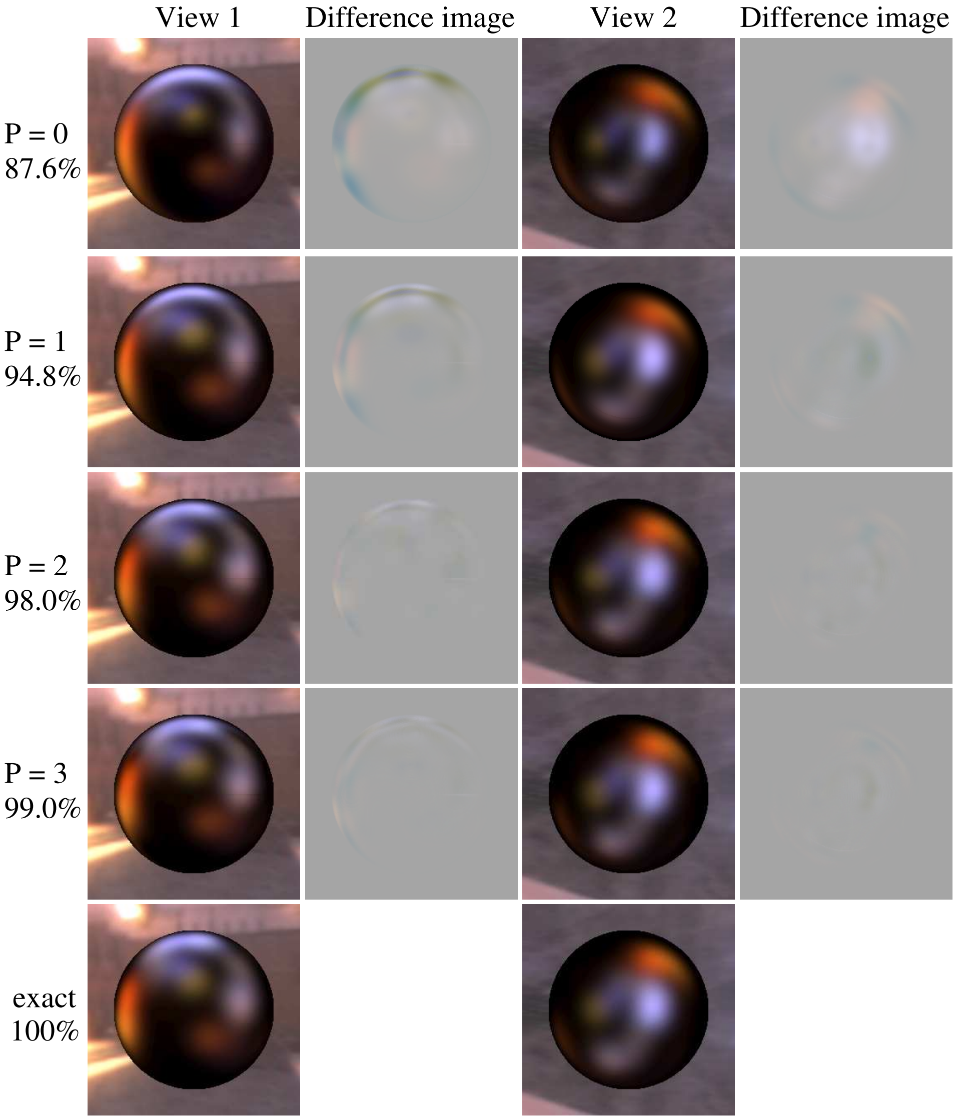

Figure 5.6: *Comparing images obtained with different values for P for a simplified microfacet BRDF model with surface roughness $\sigma = 0.2$. These images correspond to two particular views, i.e. values of (θ_o, ϕ_o). The percentages are fractions of the total energy $(1-\epsilon)$ of the BRDF captured for that P, which we use as a conservative estimate of the accuracy of the reflected light field. The exact images at the bottom were computed by a full hemispherical angular-space integral for each image pixel. For this and subsequent figures, the difference images are not amplified, and we used $F_B = 30$ and $S_B = 128$.*

Kajiya-Kay model: Finally, we tried the Kajiya-Kay [35] model, which is an anisotropic variant of the Phong BRDF, and depends on incident and outgoing angles with respect to the tangent vector. As discussed in section 5.3.1, we may reparameterize by the tangent vector, to derive $\hat{\rho} = \cos^s(\theta_i - \theta_o)$. While this paper does not consider general 4D anisotropic BRDFs, we can handle the Kajiya-Kay BRDF, since it is mathematically analogous to a (2D) isotropic BRDF after reparameterization. Unlike for ordinary Phong-like BRDFs, we cannot apply any further reflective reparameterization. Therefore, the value of P required is large (we found $P = 8$ for $s = 32$ and $\epsilon < .03$). However, there is no azimuthal dependence, so we require only $P + 1$ terms $B_{p0}(\alpha, \beta)$ in the SHRM instead of $(P + 1)^2$ (i.e. $q = 0$, with no dependence on ϕ_o). Hence, the SHRM is still a very efficient and compact representation.

5.7.2 Number of coefficients for measured BRDFs

To further evaluate the accuracy of our approximations, we used the CURET database [14]. This database consists of 61 BRDFs and BTFs, corresponding to a variety of materials. For each sample, there are 205 BRDF measurements, which may be interpolated by fitting order 8 Zernike polynomials to create a complete BRDF description [43].

Figure 5.7 is a bar chart showing, for each of the 61 samples, the accuracy of a BRDF approximation[6] with $F_{\hat{\rho}} = 30$ and values of $P_{\hat{\rho}}$ ranging from 0 to 5. In 56 cases, the accuracy for $F_{\hat{\rho}} = 30$ and $P_{\hat{\rho}} = 5$ was greater than 90% ($\epsilon < 0.1$), and was usually significantly higher (in most cases, $\epsilon < .05$ for $P_{\hat{\rho}} = 3$). The remaining 5 examples (9-frosted glass, 23-lettuce leaf, 33-slate a, 41-brick b, 57-peacock feather) were all significantly anisotropic.

Therefore, we conclude that for almost all BRDFs of interest, an order $P_B \leq 5$ suffices for the SHRM, with $F \leq 30$. In fact, for most BRDFs, a quadratic or cubic (second or third order with $P_B = 2$ or 3) spherical harmonic expansion in the SHRM suffices.

[6]We reparameterized all BRDFs by the reflection vector. Our results demonstrate that this reparameterization is suitable even if the BRDF is not primarily reflective, or consists of both diffuse and specular components. The specular components are compactly represented, while the diffuse components are low frequency anyway.

5.7.3 SHRM accuracy

We now compare images created with SHRMs to the correct image, and to previous approaches. First, figure 5.8 compares our method to Kautz and McCool's [38] 3D texture-mapping technique[7], where they approximate the BRDF—and hence, the reflected light field—as having no azimuthal dependence. For the relatively complex velvet BRDF (CURET database) in figure 5.8, their approximation introduces large errors, while the SHRM with $P_B = 5$ is accurate.

Figure 5.9 compares our approach to the correct image and Cabral's icosahedral interpolation. For sharply-varying BRDFs, such as those exhibiting strong near-grazing specularities, or complex anisotropic behavior, Cabral's approximation can lead to large errors, while our approach still gives accurate results.

In our approach, the theoretical analysis can be used to systematically trade off accuracy for compactness and efficiency. Specifically, if Kautz and McCool's [38] approximation of 2D BRDFs with no azimuthal dependence suffices ($q = 0$), we get a 3D SHRM with only $P_B + 1$ terms instead of $(P_B + 1)^2$. If Cabral et al.'s [8] icosahedral set of 12 reflection maps suffices, we can use a very small number of terms ($P_B = 1$ or 2) in the SHRM.

5.7.4 Speed of prefiltering

We first consider Phong BRDFs, experimentally validating the theoretical conclusions of section 5.6.4. For our frequency domain algorithm, we used $\epsilon = .01$, conservatively setting $F_B = 1 + \sqrt{6s}$. For the angular domain, we were more aggressive, setting $\epsilon = .05$. The resolution S_L and S_B of the inputs and final results were 128, i.e. we generated output Phong reflection maps at 128×128 resolution. According to the theory, this is an appropriate resolution for $s = 32$ and $s = 64$ (i.e. $F_B \approx 12$), and is therefore the most suitable single resolution for the entire range of Phong exponents. The numerical running times reported in table 5.4 obviously depend on our implementation and hardware. However, we believe the *ratio* in running times of angular and frequency domain methods is quite representative. Furthermore, the timing data can be fit almost precisely to the theoretical

[7] We use their single lobe model, with the BRDF being an arbitrary 2D function $\hat{\rho} = u(\theta_i, \theta_o)$. This is essentially equivalent to setting $q = 0$ in the local SHRM, using only the azimuthally independent terms.

predictions of equations 5.29 and 5.32. The results indicate that our prefiltering method is usually two orders of magnitude faster than angular-space methods, and that Phong BRDFs can be prefiltered at close to real-time rates using our approach.

Exponent s	Time (sec)		Ratio
	Angular	Frequency	(Ang/Freq)
8	67.28	0.081	830.6
16	38.03	0.114	333.6
32	21.80	0.159	137.1
64	11.94	0.227	52.6
128	7.17	0.328	21.9
256	3.55	0.461	7.7
512	2.28	0.686	3.3

Table 5.4: *Comparison of timings of angular and frequency-space prefiltering for different values of the Phong exponent s. The timings are on a 1.4GHz Pentium IV.*

As predicted by complexity analysis, the speedups are even more dramatic for the general case—illustrated using the microfacet model of equation 5.34. In table 5.5, we compare computation time for our approach and conventional methods. It can be seen that even the cost for creating the entire SHRM is much less than the cost of hemispherical integration for a single reflection map. When the cost to explicitly create multiple reflection maps is considered, our approach is three to four orders of magnitude faster.

σ	F_B, P_B	angular-space		frequency-space	
		Time (s)	Time/Image (s)	Time (s)	SHRM (s)
.1	24,3	923	9.23	2.70	1.55
.2	12,2	2744	27.44	1.55	0.72
.3	7,2	5731	57.31	1.49	0.67
.4	5,2	9034	90.34	1.47	0.65
.5	5,2	12580	125.80	1.45	0.64

Table 5.5: *Times for angular-space and our frequency-space prefiltering , with $T_B = 10$. The six columns are the value of the roughness σ, the order of expansion F_B, P_B for $\epsilon < .03$, the total angular-space computational time to create $T_B \times T_B = 100$ reflection maps, the angular-space time per reflection map, the total frequency-space time, and the frequency-space time for SHRM creation (but not explicit generation of reflection maps). Our approach is orders of magnitude faster, and even creation of the entire SHRM is usually faster than generating only a single image in angular space.*

5.7.5 Real-time rendering

There are several possibilities for real-time rendering. We could simply evaluate equation 5.10 in software for each pixel of the final image. If hardware multitexturing support is available, we may represent the spherical harmonics $Y_{pq}(\theta_o, \phi_o)$ and the local SHRM coefficients $B_{pq}(\alpha, \beta)$ by 2D texture maps. Since we are reparameterizing by the reflection vector, we will sometimes also refer to $B_{pq}(\alpha, \beta)$ as reflection maps. If $P_B = 2$, there would be 9 terms in the SHRM, corresponding to a total of 18 texture maps. We would then use graphics hardware to accumulate 9 terms, with each term being the product of two texture maps, i.e. $B_{pq}(\alpha, \beta)Y_{pq}(\theta_o, \phi_o)$. Since this algorithm is essentially that previously used for rendering factored BRDFs [37, 57], the same code can now be easily adapted for arbitrary isotropic BRDFs and complex illumination.

A simpler approach is possible when the viewer can be assumed distant, using the global SHRM in equation 5.11. The spherical harmonics $Y_{pq}(\tilde{\theta}_o, \tilde{\phi}_o)$ need be evaluated only once per frame, for given viewpoint $(\tilde{\theta}_o, \tilde{\phi}_o)$, instead of at each vertex or pixel. In fact, it is possible to render the scene using only a single reflection mapping pass. The key idea is to explicitly sum equation 5.11 to create a single dynamic 2D reflection map $\tilde{B}(\alpha, \beta)$, which is updated for every frame, i.e. each new viewpoint $(\tilde{\theta}_o, \tilde{\phi}_o)$,

$$\tilde{B}(\alpha, \beta) = \sum_{p=0}^{P_B} \sum_{q=-p}^{p} \tilde{B}_{pq}(\alpha, \beta) Y_{pq}(\tilde{\theta}_o, \tilde{\phi}_o). \tag{5.35}$$

Our implementation extends the Stanford real-time programmable shading system [68] to render with global SHRMs using equation 5.35. An advantage of our approach is that standard reflection maps can be upgraded to SHRMs with no change in the external shader programs. Internally, we simply update the reflection map for each frame. We compute equation 5.35 in software, which allows us to easily consider high-dynamic range, and avoids hardware precision and clamping issues. In the figures, the high-dynamic range backgrounds are tone-mapped, but the objects themselves are computed and shaded using a linear scale.

We used a 1.4 GHz Pentium IV running Linux, with an NVIDIA Geforce2 GTS graphics card, for our tests. The reflection (cube)map $\tilde{B}(\alpha, \beta)$ was computed at a resolution of

$64 \times 64 \times 6$, which is an appropriate resolution for most BRDFs, i.e. $F \sim 20$. Since there is real-time cube mapping hardware, the major cost is that for computing equation 5.35 in software per frame. We are able to achieve frame rates of approximately 30 frames per second, with real-time speeds even in scenes with multiple SHRMs. Figure 5.1 shows a number of examples.

5.8 Conclusions and Future Work

In this chapter, we have presented new frequency-space algorithms for real-time rendering of complex isotropic BRDFs under arbitrary distant illumination, and have validated our approach using many different BRDFs and lighting conditions. Our contributions include theoretical analysis that allows us to precisely determine the orders of our spherical harmonic expansions, the new compact and efficient SHRM representation for the reflected light field, and very fast prefiltering algorithms based on spherical harmonic transforms. We have integrated the three contributions into a complete frequency-space pipeline, as per figure 5.2. However, it is also easy to convert between SHRMs and previous explicit representations. Therefore, the contributions of this paper are relatively independent, and can also be incorporated separately.

There are several interesting similarities and differences between the SHRM and surface light field representations. In fact, the SHRM can be seen as a surface light field on a sphere. The main advantage of the SHRM is that it is independent of geometry, i.e. it can be mapped on to any object geometry using the surface normal.

On the other hand, surface light fields have a number of advantages over SHRMs. They capture the effects of spatially varying illumination, which includes non-uniform lighting, as well as geometry-specific interreflection and self-shadowing effects.

Another difference between SHRMS and surface light fields is that our method is primarily a synthetic approach, while surface light fields represent collected data. Of course, it is possible to convert a surface light field on a sphere to an SHRM, or to synthesize a surface light field given the geometry, reflectance and illumination.

Since surface light fields are so large, they must be compressed. It has been widely assumed that the optimal compression algorithm for light fields will be based on model-based prediction and coding. SHRMs expose the structure of the reflected light field in terms of the frequency properties of the illumination and BRDF. It is therefore interesting to compare spherical harmonic basis functions to PCA-based compression and factorization methods used for BRDFs [37] and surface light fields [64, 88].

The main advantage of SHRMs is that the theoretical analysis gives insight into the intrinsic complexity of the reflected light field, given the complexities of the illumination and reflection functions. This allows us to directly compute the SHRM with the right order and resolution. Furthermore, our approach is much more efficient than computing a full PCA on the entire reflected light field. This is because brute-force PCA computation requires a dense 4D reflected light field as input, and requires computing a singular-value decomposition, which is an expensive operation. Our approach, in contrast, only computes the terms that are needed and uses very fast prefiltering methods.

On the other hand, our method uses a predetermined basis—the spherical harmonics, while PCA-based methods have the advantage of finding an optimal basis to represent a data set, assuming no a priori knowledge. As future work, it would be interesting to see if PCA can be run directly on the SHRM in order to compute a compact final representation efficiently, thus getting the best features of both PCA, and our SHRM representation.

Several aspects of this paper are likely to have broader implications in the general context of rendering. SHRMs are related to Sillion et al.'s [79] spherical harmonic representation of outgoing light for radiosity computations, but differ in storing B_{pq} as a function of orientation (α, β), rather than at each vertex of a geometric model. Because we reparameterize by the reflection vector, our representation is much more compact. SHRMs are also similar in spirit to the representation of Malzbender et al. [53]. Like us, they have a 4D function with rapid variation over two dimensions (for them, texture), and slow variation over the other two (for them, illumination). Since spherical harmonics are polynomials of the cartesian components, their (quadratic) polynomial texture maps can be seen as a subset of our representation.

One drawback of synthetic IBR is the long time required for precomputation, which precludes dynamic lighting or interactive manipulation of material properties. Our new prefiltering method takes an important step in addressing this problem for environment maps. However, we do not currently take into account near-field lighting, interreflection or self-shadowing. In the future, we wish to address these limitations, adapting our prefiltering method to rapidly compute synthetic surface light fields.

For the special case of radially symmetric BRDFs, Kautz et al. [40] have proposed using hardware-assisted 2D image convolution. However, while BRDFs are shift-invariant filters on the spherical domain, they are not shift-invariant in the plane, since projection on to a 2D image introduces distortion [40], and may lead to inconsistencies—for instance, rotating the lighting may not correspond simply to rotating the prefiltered image. Our prefiltering algorithm can be viewed as spherical image processing on the incident illumination, convolving it with the BRDF filter. Our speedups are not surprising, given that planar image convolutions are often more efficiently computed in the Fourier domain. Other approaches to speed up prefiltering are hierarchical methods [40] and spherical wavelets [78]. However, *there is no wavelet or hierarchical convolution formula*, so frequency domain methods are more appropriate for environment mapping. and will give asymptotically better results according to our computational complexity analysis. Further, hierarchical methods can be seen as effectively determining angular resolutions appropriately for a given error tolerance, which we are able to formally analyze.

There has been considerable recent interest in real-time rendering with complex illumination and material models. In work simultaneous with ours, Latta and Kolb [47] have used homomorphic factorization to represent the reflected light field as aa product of textures. As pointed out in the text, the SHRM may be seen as a sum of a number of terms, each being the product of factors depending on the reflected and viewing directions. Having a small number of terms, determined by error analysis, allows greater accuracy. A further significant advantage of the SHRM is that it can be computed extremely efficiently. On the other hand, Latta and Kolb's approach [47] of factoring after computing a dense reflected light field representation has the advantage of allowing for a number of different factorizations, such as using the surface normal instead of the viewing direction, which may be more suitable for diffuse BRDFs. Also, simultaneous with our work, Sloan et al. [66] have restricted themselves to low-frequency lighting, but shown how complex light transport such as shadows and interreflection may be precomputed and then rendered in real-time.

The SHRM differs in that we consider natural illumination, which has significant high-frequency content, but assume no shadowing or interreflection.

In the future, we would like to combine and extend the functionality of these different approaches, so we may handle general near-field illumination, spatially varying realistic materials, and physically-correct light transport including shadows and interreflection. We would also like to adapt the methods of this paper, such as the sampling rate analysis and the SHRM representation, to other image-based rendering problems in graphics.

In summary, natural illumination and accurate BRDFs are of growing importance in interactive applications, and this chapter has presented a complete frequency-space pipeline to enable this.

In the these last two chapters, we have seen the application of the signal-processing framework to problems in interactive forward rendering. The next chapter applies the framework to inverse rendering—estimating illumination and material properties from images.

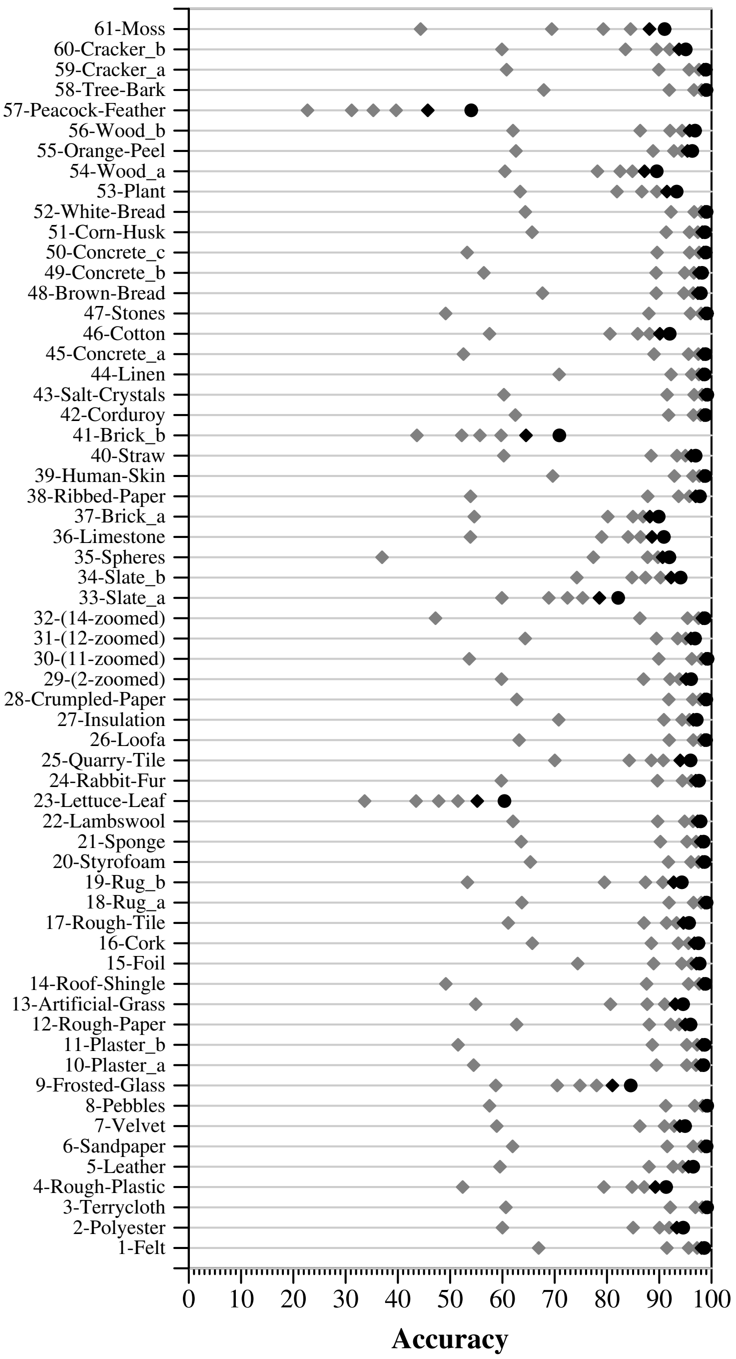

Figure 5.7: *Accuracy of a spherical harmonic BRDF approximation for all 61 BRDFs in the CURET database. We show 6 values of $P_{\hat{\rho}}$ ranging from 0 to 5 from left to right. The low orders for $P_{\hat{\rho}}$ are shown with light gray diamonds, while a black circle shows the highest order $P_{\hat{\rho}} = 5$. Note that the rightmost circle corresponds to an accuracy greater than 90% in 56 of the 61 rows.*

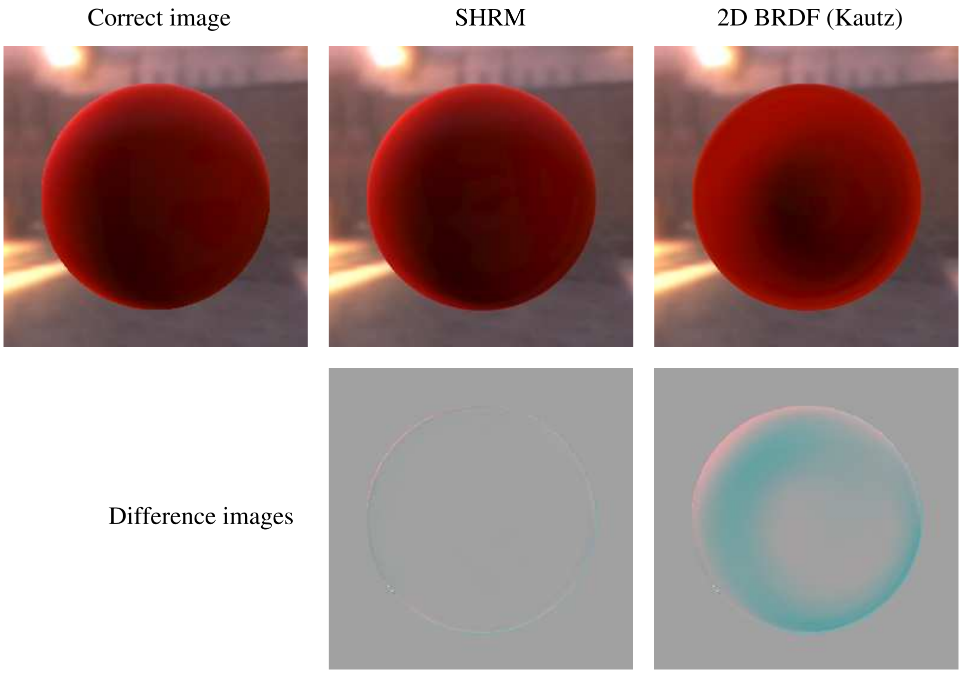

Figure 5.8: *Comparing the correct image on the left to those created using SHRMs (middle) and the 2D BRDF approximation of Kautz and McCool (right).*

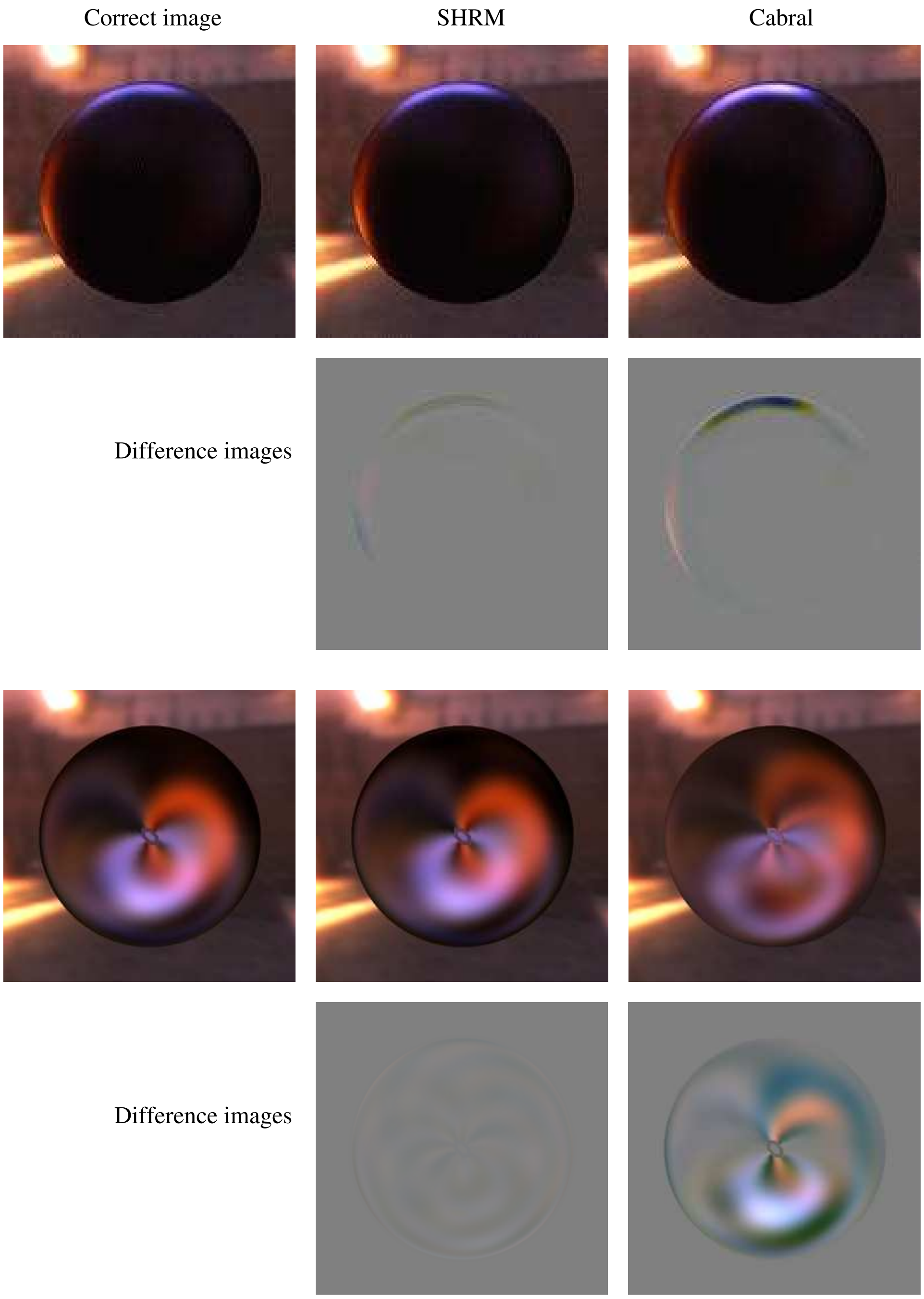

Figure 5.9: *Comparing the correct image to those created using SHRMs and icosahedral interpolation (Cabral's method). We see that the SHRM image is accurate, while Cabral's approximation is inadequate for sharp near-grazing reflections (top), and for complex BRDFs like the anisotropic Kajiya-Kay model (bottom).*