Foundations and Trends® in
Computer Graphics and Vision
Vol. 3, No. 4 (2007) 281–369

DOI: 10.1561/0600000021

now
the essence of knowledge

# Precomputation-Based Rendering

By Ravi Ramamoorthi

## Contents

**1 Introduction** **282**

**2 Background and Basic Precomputation Method** **287**

2.1 Reflection Equation 287
2.2 Rendering Equation 289
2.3 Precomputation-Based Image Relighting 291

**3 Environment Maps** **295**

3.1 Reflection and Environment Mapping 296
3.2 Reflection as Convolution 300
3.3 Spherical Harmonics 302

**4 Precomputed Radiance Transfer** **307**

4.1 Challenges and Assumptions 308
4.2 Precomputed Radiance Transfer Algorithm 311

**5 Basis Representations and Compression** **318**

5.1 Overview: Relighting as Matrix-Vector Multiplication 318
5.2 All-Frequency Relighting: Wavelet Lighting Approx. 320
5.3 Transport Compression: Clustered Principal Components 324

**6 Dynamic Lighting, View, Materials, Geometry** **329**

6.1 All-Frequency Relighting with Changing View 329
6.2 Dynamic Materials and BRDF Editing 338
6.3 Dynamic Scenes and Deforming Geometry 341
6.4 Discussion and Limitations 347

**7 Extensions** **349**

**8 Future Research** **353**

**9 Conclusion** **356**

**Acknowledgments** **358**

**References** **360**

Foundations and Trends® in
Computer Graphics and Vision
Vol. 3, No. 4 (2007) 281–369

DOI: 10.1561/0600000021

now
the essence of knowledge

# Precomputation-Based Rendering

**Ravi Ramamoorthi**

*University of California at Berkeley, 627 Soda Hall, Berkeley, CA 94720-1776, USA, ravir@cs.berkeley.edu*

## Abstract

High quality image synthesis is a long-standing goal in computer graphics. Complex lighting, reflection, shadow and global illumination effects can be rendered with modern image synthesis algorithms, but those methods are focused on offline computation of a single image. They are far from interactive, and the image must be recomputed from scratch when any aspect of the scene changes. On the other hand, real-time rendering often fixes the object geometry and other attributes, such as relighting a static image for lighting design. In these cases, the final image or rendering is a linear combination of basis images or radiance distributions due to individual lights. We can therefore *precompute* offline solutions to each individual light or lighting basis function, combining them efficiently for real-time image synthesis. *Precomputation-based relighting and radiance transfer* has a long history with a spurt of renewed interest, including adoption in commercial video games, due to recent mathematical developments and hardware advances. In this survey, we describe the mathematical foundations, history, current research and future directions for precomputation-based rendering.

# 1

# Introduction

High quality image synthesis is one of the oldest goals of computer graphics. A standard to aspire for is often referred to as *photorealism* — rendering images indistinguishable from real photographs. Achieving this goal requires considering a variety of complicated shading effects in the real world, such as complex natural illumination from a skylit scene, soft shadows from the leaves of a tree in sunlight, glossy reflections from a velvet cushion, and caustics from a wineglass. Three decades of research in offline global illumination algorithms has enabled substantial progress towards these goals, and a variety of complex lighting, reflection, shadow and global illumination effects can be rendered. The evidence is for all to see in the form of completely computer-generated movies, or the increasing use of computer graphics in the movie industry to seamlessly combine live action and synthetic elements.

There is another class of applications however, that requires real-time performance. Examples include video games, lighting design for architects and animators, interactive simulation and training, visualization of artifacts for archaeology and e-commerce, and virtual worlds. Historically, there has been a large chasm between interactivity and

realism. While offline computer graphics rendering achieved more and more realistic effects, real-time imagery focused on increasing raw performance for geometry and textures. Interactive applications usually did not include complex natural lighting, realistic materials or accurate shading — indeed, cast shadows and global illumination were often completely missing in real-time rendering.

Over the past decade, serious efforts began to be made to bridge this chasm between photorealism and real-time. Two critical developments in computing power played a key role. First, graphics hardware became increasingly fast and flexible. Over a number of years, a new brand of graphics hardware and graphics processing units or GPUs evolved [16, 84], along with associated programming languages [17, 90, 105]. This enabled complex shading models and physically realistic computations to be performed at each vertex or pixel. The focus thus shifted from raw performance to high quality real-time image synthesis; an early paper in this direction is by Heidrich and Seidel [56], and a survey on this topic is by Kautz [67]. A second important development was the retargetting of traditional global illumination and ray tracing methods to modern CPU and GPU hardware. More efficient algorithms adapted to modern hardware, coupled with several iterations of Moore's law, enabled the first methods for real-time ray tracing [54, 106, 114, 131, 135, 136, 149]. A good recent survey of the work on the topic, including animated scenes, is the STAR report by Wald et al. [137].

However, many complex shading effects still proved quite difficult to address at real-time rates. A key challenge is the complexity of illumination. Real scenes are lit, not by a single point light source, but by multiple lights including area sources (and even virtual lights for instant global illumination [73]). Moreover, over the last decade, there has been substantial interest in high-dynamic range representations of full incident illumination, known as an *environment map* [15, 28, 93], where each of the million or more pixels can be viewed as a light source. Furthermore, area and environment lighting create a new type of look, involving more diffuse shading and softer shadows, that is often a desirable mood. However, rendering with multiple lights involves adding up or integrating their contributions. Even if we had an interactive method for a single light source, it will become very slow once we

consider the hundreds to thousands of lights needed for realistic incident illumination.

This survey is about a class of techniques that use precomputation to address the challenge of complex illumination. They are collectively referred to as *precomputation-based relighting* or *precomputed radiance transfer* (*PRT*). The key idea is that the final radiance distribution in the scene is linear in the individual light sources — we can first simply render or *precompute* the results offline for each light source or lighting basis function, and then rapidly sum up the results in a second real-time phase. Of course, doing this efficiently, especially when there are thousands of basis lights as in an environment map, is non-trivial and an important intellectual challenge.

It is important to note the key distinction between offline rendering and PRT. Traditional global illumination takes as input the geometry, view, lighting and object material properties, and can produce very realistic images, albeit slowly. However, the algorithm must be re-run completely when any of these attributes change. In real-time rendering applications, we must be able to update the image interactively when, for example, illumination changes, but can often assume that other properties like object geometry remain fixed. Indeed, a precomputation-based approach does require fixing certain scene attributes. The earliest techniques allowed only the lighting to be dynamic, with viewpoint, geometry and materials all fixed. These are essentially approaches for *image relighting*, and the initial seminal work in this area by Nimeroff et al. [99], Dobashi et al. [34], and Dorsey et al. [35] was motivated by applications like a time-sequence of a scene lit by natural illumination, and the interactive design of operatic stage lighting.

Starting in 2002, with the publication of a seminal paper by Sloan et al. [121], the term "*precomputed radiance transfer* (*PRT*)" became common in computer graphics and a topic of renewed interest. One key innovation was the ability to address complex broad-area environment lighting, based on spherical harmonic representations inspired by a theoretical result the previous year on reflection as convolution [11, 108, 110]. A second important idea was to represent the radiance distribution on geometric meshes, enabling the idea to progress from 2D image relighting to rendering on 3D graphics hardware.

Fig. 1.1 Some examples of the types of renderings produced by Precomputed Radiance Transfer systems. The lighting can be varied dynamically with a variety of complex reflection and shadowing effects rendered in real-time. The image on the left is from the seminal initial work of Sloan et al. [121], while the image on the right is for relighting from a detailed illumination cubemap while preserving shadow and other features at all frequencies [96]. To enable precomputation, the geometry of the scene is assumed fixed in these examples.

Example images from early PRT algorithms [96, 121] are shown in Figure 1.1.

Since its inception, precomputed radiance transfer has led to a large number of papers as well as commercial adoption, with a number of video game companies (e.g., Microsoft and Bungie) incorporating variants, and a version included in Microsoft's DirectX 9 Utility Library. It has also led to a variety of new theoretical insights on basis representations and decompositions and analyses of light transport, that are broadly applicable to other domains as well. In this survey, we present a unified mathematical view of precomputation-based rendering, while discussing its motivation, history, and current and future research directions. Advanced readers may also be interested in a deeper technical discussion of a framework for precomputed and captured light transport by Lehtinen [79].

The remainder of this survey is organized as follows. Section 2 discusses background on the rendering equation and early seminal work in precomputation-based image relighting. Section 3 describes early and recent work on environment maps, a spherical distribution of the

lighting in the scene. We also discuss key theoretical results developed in 2001, that showed that an effective representation of light reflection could be achieved using spherical harmonics. Section 4 introduces the 2002 paper by Sloan et al. [121] that has spurred much of the recent work in PRT. Section 5 describes different types of basis representations and compressions that have since been applied to precomputed light transport. Section 6 discusses the relaxation of a number of restrictions including varying viewpoint, materials and dynamic geometry. Section 7 describes some of the recent work on variants of PRT, that address issues like global illumination, lighting design, and volumetric scattering. Following this, Section 8 discusses future research directions. We conclude the survey in Section 9.

# 2

# Background and Basic Precomputation Method

In this section, we introduce some background material on the reflection and rendering equations, that most precomputation-based methods attempt to solve. We then briefly describe some of the earliest seminal work in precomputation-based relighting by Nimeroff et al. [99], Dobashi et al. [34], Dorsey et al. [35], and Teo et al. [132].

## 2.1 Reflection Equation

The goals of realistic image synthesis can be seen as solving approximations to an equation known in computer graphics as the rendering equation, first introduced by Kajiya [64]. To develop that framework, we start with the simpler reflection equation [24], that is often used as the basis for modern precomputed radiance transfer systems:

$$B(\mathbf{x},\boldsymbol{\omega}_o) = E(\mathbf{x},\boldsymbol{\omega}_o) + \int_{\Omega_{2\pi}} \rho(x,\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) L_i(\mathbf{x},\boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \mathbf{n})\, d\boldsymbol{\omega}_i, \quad (2.1)$$

where $B(\mathbf{x},\boldsymbol{\omega}_o)$ is the reflected or outgoing radiance, corresponding to the image we want to render (the notation $B$ is used more commonly in current PRT literature; $L_r$ is an alternative). $E$ is the emission, corresponding to light sources only (an alternative notation is $L_e$, since

$E$ is sometimes used to denote irradiance). The variables $\mathbf{x}$, $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ are respectively, the spatial location, incident and outgoing angles (since we are in 3D, they are vectors).

The integral on the right of Equation (2.1) is over the visible or upper hemisphere $\Omega_{2\pi}$. The integrand involves the BRDF [98] or Bi-Directional Reflectance Distribution Function $\rho$, that controls the way the surface reflects light and distinguishes materials like plastic, paints or wood. Note that incident and outgoing angles $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$, are in the local coordinate frame of the surface, with respect to its normal. $L_i(\mathbf{x},\boldsymbol{\omega}_i)$ is the incident illumination at every point $\mathbf{x}$ from incident directions $\boldsymbol{\omega}_i$. Where obvious from context, we sometimes drop the subscript and simply use $L$. Finally, $(\boldsymbol{\omega}_i \cdot \mathbf{n})$ is the cosine of the incident angle, or dot product between the normal and incident direction. The reflected light is an integral or summation over illumination from all incident directions, weighted by the BRDF of the surface, and the cosine of the incident angle. A table of notation for our discussion of the reflection and rendering equations is in Table 2.1.

In current PRT systems, it is common to specify $L_i$ as illumination from an *environment map* [15, 46, 93], which is simply a representation of the incident illumination in the scene. The idea and technique are

Table 2.1 Table of basic notation for reflection and rendering equations.

| | |
|---|---|
| $B$ | Reflected or outgoing radiance |
| $E$ | Emissive radiance (from light sources) |
| $L$ or $L_i$ | Incident radiance |
| $\rho$ | Surface BRDF |
| $\mathbf{x},\mathbf{y}$ | Spatial locations (in 3D) |
| $(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)$ | Incident and outgoing angles |
| $\mathbf{n}$ | Surface normal |
| $G(\mathbf{x},\mathbf{y})$ | Geometric factor in rendering equation |
| $V(\mathbf{x},\mathbf{y})$ | Visibility between $\mathbf{x}$ and $\mathbf{y}$ |
| $\mathbf{K}$ | Local reflection operator |
| $\mathbf{G}$ | Geometric operator (outgoing to incident radiance) |
| $\mathbf{T}$ | Light transport operator $\mathbf{T} = (\mathbf{I} - \mathbf{KG})^{-1}$ |
| $\alpha_j$ | Coefficients of lighting in some basis |
| $Y_{lm}$ | Spherical harmonic (Section 3.3) |
| $d\boldsymbol{\omega}$ | Differential solid angle |
| $\Omega$ | (Hemi)sphere of integration |
| $dA$ | Differential area |
| $S$ | Surfaces in scene for integration |

very old, and can be traced back to Blinn's initial work on reflection mapping in 1976 [15], and the subsequent seminal developments of Miller and Hoffman [93]. With the recent popularity of high-dynamic range photography, Debevec and collaborators [28, 30] have spurred a great deal of renewed interest in environment mapping over the last decade. The use of this type of detailed photographed natural illumination has enabled modern PRT techniques, and is the topic of Section 3.

A common assumption in environment mapping is that the illumination field is distant, so that $L_i(\mathbf{x}, \boldsymbol{\omega}_i) = L_i(\boldsymbol{\omega}_i)$. (Note that we are not considering local occlusions from the object itself here, but the (global) illumination field at each point.) The lighting $L_i(\boldsymbol{\omega}_i)$ can now be acquired as a 2D image, which is most commonly done by photographing a chrome steel or mirror sphere [93], that simply reflects the incident lighting. This lighting environment $L_i$ can then be rotated or otherwise transformed dynamically, with real-time updates to the reflected light or image $B$. The challenge that PRT algorithms must solve, is how to integrate over all incident directions, or add up lighting contributions from every pixel in the environment map in real-time.

One critical aspect of Equation (2.1) is important to note. The equation for $B$ is "linear" in the incident illumination $L_i$. That is, if $L_i$ is composed of two components $L_i = aL_i^1 + cL_i^2$, the result can also be linearly composed, $B = aB^1 + cB^2$. This "linearity of light transport" is a key enabling factor in PRT algorithms.

## 2.2 Rendering Equation

While the reflection equation, especially using environment maps and PRT algorithms, can produce some stunning results, it still allows only for direct lighting that comes directly from light sources or a distant environment map. Fully realistic image synthesis requires multiple bounces of light or interreflection between objects, and can be addressed generally using the rendering equation [64],

$$B(\mathbf{x}, \boldsymbol{\omega}_o) = E(\mathbf{x}, \boldsymbol{\omega}_o) + \int_S \rho(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) B(\mathbf{y}, -\boldsymbol{\omega}_i) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y})\, dA. \tag{2.2}$$

The incident illumination in the integral now corresponds to the reflected light $B(\mathbf{y}, -\boldsymbol{\omega}_i)$ from point $\mathbf{y}$. Here, $\mathbf{y}$ corresponds to the set of points in the direction $\boldsymbol{\omega}_i$ from $\mathbf{x}$, with the closest such point being picked by the visibility function $V(\mathbf{x},\mathbf{y})$. The integral is now over $S$, the area of surfaces in the scene (hence $dA$), and includes a geometric term $G(\mathbf{x},\mathbf{y})$ that depends on the orientations and distances of $\mathbf{x}$ and $\mathbf{y}$, as well as a binary visibility term $V(\mathbf{x},\mathbf{y})$ that controls if $\mathbf{x}$ and $\mathbf{y}$ can see each other.

The structure of this integral is quite complicated. Following the work of Arvo [7], one can gain insight into its form by writing in operator notation,

$$B = E + \mathbf{K}(\rho)\mathbf{G}B, \tag{2.3}$$

where $\mathbf{G}$ is a geometric operator that takes outgoing or reflected radiance and propagates it within the scene to obtain incident radiance, while $\mathbf{K}$ is a local linear reflection operator that takes incident radiance and turns it into reflected light, as illustrated in Figure 2.1. Following Ben-Artzi et al. [13], we note explicitly that $\mathbf{K}(\rho)$ depends on the BRDFs in the scene — in fact, it is a bilinear operator in both the BRDFs and the scene lighting.

It may help to explicitly expand Equation (2.3),

$$B = (\mathbf{I} - \mathbf{KG})^{-1}E = E + \mathbf{KG}E + \mathbf{KGKG}E + \cdots \tag{2.4}$$

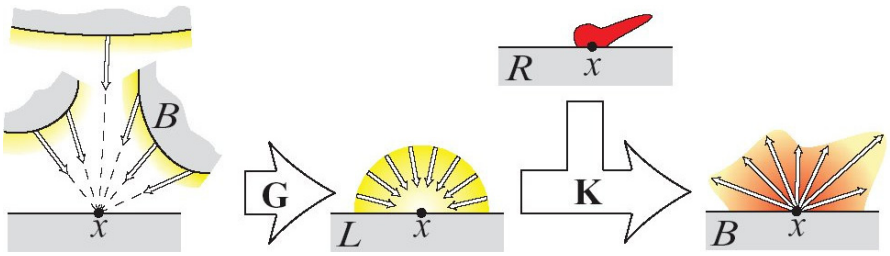


Fig. 2.1 Schematic of the rendering equation (taken from [13]). Outgoing radiance from all surfaces in the scene (left) is converted by the $\mathbf{G}$ operator into local incident radiance at each surface location (middle), which is then reflected by the bilinear local reflection operator $\mathbf{K}$. This operator is bilinear, taking as inputs both the incident lighting and the BRDF. Finally, we obtain the reflected light $B$ (right).

where on the right-hand side each term in the series refers to the corresponding number of bounces of light. There are two main approaches to compute the reflected light $B$, or numerically solve the rendering equation. They correspond to ray tracing [62, 147] or Monte Carlo methods, and radiosity [24, 41] or finite element methods. In this survey, we will not usually focus on the details of how the rendering (or reflection) equation is solved, since that is a precomputation in PRT methods. There is a large literature on the subject, now spanning almost three decades, and we will assume there exists a good rendering engine for the precomputation phase.

A key point is that while the integral is now more complicated than that for the reflection equation, the final result $B$ is still "linear" in the initial lighting distribution or emission $E$. In fact, it is possible to precompute a net transport operator $\mathbf{T} = (\mathbf{I} - \mathbf{KG})^{-1}$, and then calculate $B = \mathbf{T}E$; this has effectively been the approach taken in much recent work on PRT with interreflections [52, 77].

In summary, we have introduced the reflection and rendering equations, that form the foundations for all of the work on precomputation-based methods. A key property is the "linearity of light transport" — the final image $B$ is linear in the incident illumination. Precomputation-based methods exploit this by precomputing the final result for each lighting direction, and then linearly combining these results at run-time based on the actual dynamic illumination.

## 2.3 Precomputation-Based Image Relighting

Having given the basic background, we now review the earliest seminal works on precomputation-based image relighting. An initial paper is by Nimeroff et al. [99] for quickly re-rendering scenes under changing natural illumination conditions. Much of the tone for subsequent work is set in this paper, as the authors note that "*Re-rendering is accomplished via linear combination of a set of pre-rendered* '*basis*' *images.*"

To understand Nimeroff et al. [99] and much of the more recent work in PRT, consider Equation (2.1), assuming incident light from a distant environment map — nearly identical equations can be written for the full rendering equation in Equation (2.2) and other scenarios.

Consider a fixed view (image) and lighting from a given illumination condition $L_j(\boldsymbol{\omega}_i)$. Then,

$$B_j = \mathbf{T} L_j, \tag{2.5}$$

where $\mathbf{T}$ is the net light transport or rendering operator, and $B_j$ is the image that would result from the light $L_j$. The function $B_j$ can be "precomputed" offline simply by rendering an image with lighting $L_j$.

Assume we have precomputed $B_j$ for some set of functions $L_j(\boldsymbol{\omega}_i)$. The final illumination $L$ can then be approximated

$$L(\boldsymbol{\omega}_i) \approx \sum_{j=1}^{N} \alpha_j L_j(\boldsymbol{\omega}_i), \tag{2.6}$$

as a linear combination of $N$ basis functions $L_j(\boldsymbol{\omega}_i)$, with weights $\alpha_j$. The approximation will be exact when the basis functions can accurately reproduce the incident illumination with a linear combination. Choosing the appropriate basis functions for a good approximation is one of the key research areas in PRT.

By the linearity of light transport, the final image can also be computed as a similar linear combination,

$$B \approx \sum_{j=1}^{N} \alpha_j B_j, \tag{2.7}$$

where it should be emphasized that the $\alpha_j$ are the same coefficients as in the lighting expansion in Equation (2.6) and are already known. Note that this calculation can be done very fast if the number of basis functions $N$ is small, since it just involves weighting and summing the basis images $B_j$, that have already been precomputed. This is the key point of precomputation-based relighting methods — the actual runtime computation can be very efficient and often real-time, even if it takes hours to compute the individual basis images, or to compute the final rendering from scratch.

One of the main choices to be made in this approach is what and how many basis functions to use. Nimeroff et al. [99] introduce the notion of steerable functions, based on the work of Freeman and Adelson [36]. As they note, "A steerable function is one that can be written as a

linear combination of rotated versions of itself." They use a basis made of low-order polynomials over the sphere of incoming directions, using a total of nine terms (constant, linear, and quadratic polynomials). They also describe special functional forms for a variety of models of skylight. In fact, the basis they use is very similar to the spherical harmonics [34, 86] upto order 2, that we will discuss in more detail in the next section. We also show there that nine terms or order 2 spherical harmonics capture most of the energy for diffuse Lambertian shading (but not for more complex effects like specularities and cast shadows).

In subsequent work, Dobashi et al. [34] used spherical harmonic basis functions for the directional distribution of light source intensity, fixing their positions. This application demonstrates the flexibility of the precomputation-based framework. Although, we are no longer discussing environment maps or distant lighting, linearity of illumination is a flexible concept, broadly applicable to sum lighting distributions.

Basis functions like spherical harmonics require a large number of terms to capture high-frequency directional effects like sharp shadows or glossy highlights. Therefore, to come close to the quality of a direct rendering, we must still sum a very large number of basis functions $N$. While this may be faster than direct rendering, it still makes the precomputation-based approach slow for high fidelity images. This in turn leads to thinking about a number of *compression schemes*.

One key element of compression or dimensionality reduction was well appreciated by the early pioneers. Dorsey et al. [35] used a singular-value decomposition (SVD) [40] to compress and reduce the number of basis images needed in design of time-dependent lighting. This is a standard compression/dimensionality reduction method, that has been applied with great effect to many problems in different fields. Later, Teo et al. [132] proposed a general framework to handle many different types of light sources with steerable functions. They stacked the basis images $B_j$ in Equation (2.5) into a large matrix $M$, where a single column $j$ of $M$ is a linearized version of basis image $B_j$. By performing an SVD on the matrix $M$, a reduced set of "key" basis images can be obtained, that can lead to significant efficiencies in on-line computation and storage.

In summary, linearity of light transport enables fast methods for precomputation-based image relighting. All precomputation schemes need to make two key design choices — the basis functions to be used, and the compression scheme (and/or choice of the number of basis functions $N$). Spherical harmonics and steerable basis functions, along with SVD/PCA compression, are an important initial step. However, recent years have shown that new bases can provide benefits in some situations, and novel compression and factorization methods can produce significant benefits over a standard SVD. A more detailed discussion of these and other choices for basis functions and compression methods, will follow in later sections when we survey modern PRT methods.

# 3

# Environment Maps

Much of the recent popularity of precomputation-based relighting owes to the availability of *environment maps* — an image-based representation of the incident lighting. Environment maps enable rich natural illumination environments to be used, instead of point sources. Indeed, recent work in perception has shown that humans can perceive material properties much better under a natural environment map than a point source. However, their applicability in the interactive setting remains a challenge, because of the need to integrate or add up contributions over all lighting directions (or environment map pixels). The realism enabled by environment maps in real-time applications, has spurred much of the recent work on precomputation-based methods.

In this section, we review the history of reflection and environment maps, and modern developments therein. We start by discussing the early seminal work in the mid-1970s and 1980s [15, 46, 93], followed by the more recent work in the mid-1990s on high-dynamic range imaging and general reflection models [19, 28, 30, 72]. In the second half of this section, we describe developments in 2001, that introduced a new theory of reflection as convolution [11, 108, 110], and showed

how environment maps could be effectively used for real-time rendering. These papers inspired the development of the initial precomputed radiance transfer method [121] in 2002, which is described in Section 4.

## 3.1 Reflection and Environment Mapping

This section describes the early work on the reflection and environment mapping technique. The reader is particularly referred to the excellent history compiled by Debevec [31], from which we draw heavily.

In the early years of computer graphics, the only available light sources were point or directional sources (and later area lights). Therefore, reflections from an object could only consist of single glossy highlights. There was no way, for example, to model the reflection of a window with its rich outdoor natural illumination on a shiny teapot. Blinn and Newell [15] introduced reflection mapping to enable these visual effects (see Figure 3.1). They used a hand-painted reflection map to represent the incident light from the environment — these can be considered the first environment maps in computer graphics. Given a shiny object, one can easily compute the reflection of the viewing direction about the surface normal for each pixel. This reflected direction is then used to index into the reflection map to get the shading.

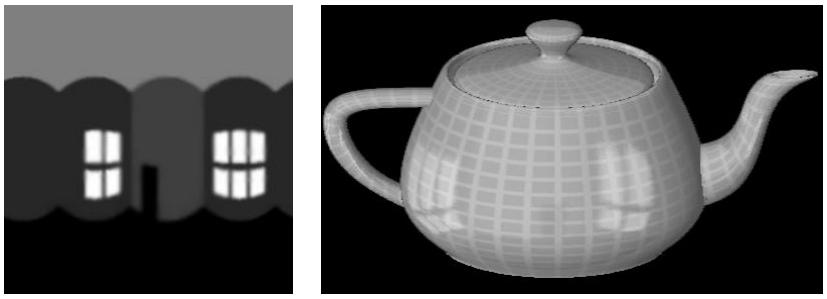

Fig. 3.1 The earliest example of reflection mapping, from Blinn and Newell [15]. The image on the left is of a room with windows, created in a paint program, and can be considered one of the first environment maps in computer graphics. By using the reflected direction to index into it, we obtain the reflection-mapped teapot on the right (note that the teapot is also lit by a point light source for the diffuse shading).

Fig. 3.2 Environment mapping, from Miller and Hoffman [93]. From the photograph of a chrome steel or mirror sphere (the Christmas lawn ornament on the left), the environment map can be obtained. Synthetic objects, like the robotic dog on the right, can then be inserted in real scenes as if they were actually present, with realistic lighting and reflections.

The idea of reflection mapping opens up the possibility of taking an omnidirectional photograph of a real environment, and using this real *environment map* as the reflection map, as seen in Figure 3.2. Within the context of Equation (2.1), the environment map is simply $L_i(\mathbf{x}, \boldsymbol{\omega}_i)$. While this is strictly valid only at a single spatial location $\mathbf{x}$, and a new environment map would need to be used elsewhere, a typical assumption is that the lighting is distant. Thus, an environment map is typically a single function $L_i(\boldsymbol{\omega}_i)$ used at all spatial locations. Extensions to multiple layers for more near-field effects have also been explored [50]. In terms of representation, environment maps are functions over the sphere of incident directions. The simplest parameterization is a rectangular latitude–longitude map, i.e., just the $(\theta, \phi)$ spherical coordinates. Obviously, this leads to large distortions, wherein the poles are heavily oversampled while the equator is undersampled. Therefore, a variety of other options have been explored in computer graphics and other fields. One very popular current representation is a "cubemap," as shown in Figure 3.3 right. The sphere is projected onto the six faces of a cube, that are then unwrapped. The cubemap is both very simple for hardware implementation, and has relatively fewer distortions, compared to a latitude–longitude map.

Fig. 3.3 *Left*: The animation "Interface" from 1985, shows a young woman kissing a robot. The scene was filmed with the actress kissing a 10-inch gazing ball, and the robot synthetically rendered later and composited in using this environment map. *Right*: A cubemap representation of an environment map, wherein the sphere of directions is projected onto the six faces of a cube, that are then unwrapped. The simple cubemap representation is widely used in graphics hardware, and in most modern PRT algorithms.

The environment can be acquired either by using a camera with a fisheye lens, or more commonly by taking one or more photographs of a light probe, typically a chrome steel or mirror sphere — a Christmas lawn ornament was one of the first examples of this light probe; current researchers typically use more than one image of a high grade mirror sphere available from a number of commercial manufacturers. As noted in [31], the environment mapping technique was developed independently by Miller and Perlin, as well as Chou and Williams. The initial uses were to introduce synthetic objects into real scenes with realistic lighting (see Figure 3.2 for early examples).

An early seminal paper on environment mapping is by Miller and Hoffman [93]. This work has sometimes not been given due credit in more recent papers, as it appears only in the SIGGRAPH 1984 course notes (that are not widely available; however, see the citation [93] for a current URL). Well before its time, this work discusses pre-convolution of the environment map to generate diffuse and specular reflection maps, the storage of an environment map as perspective images on six cube faces, and issues arising from limited dynamic range of film and how they could be addressed by using multiple exposures. In many ways, this paper previews and predicts the key developments in the late

1990s and early 2000s, that made environment mapping a very practical and widely available tool in interactive and offline applications (now used in almost any video game or movie).

Subsequent early papers on environment mapping included work by Greene [46] in 1986, who formalized many of the techniques therein, and introduced the use of cube maps, with summed area tables [25] for anti-aliasing. The initial environment mapping papers were all intended for offline use, but advances in graphics hardware enabled them to enter the real-time domain [48]. Environment mapping also has a long history in movie production. One of the earliest films was Interface in 1985, with a young woman kissing a robot (see Figure 3.3). The actress actually kissed a 10 inch shiny ball, and the reflection from the ball was used to infer the environment map, with which the robot was synthetically rendered and composited. Subsequently, the technique found its way into feature films, including Flight of the Navigator (1986), Abyss (1989), and Terminator 2 (1991).

After the seminal early work in the mid-1980s, a second leap in environment mapping was made in the late-1990s owing to the work of Debevec and collaborators [28, 30]. The first contribution was the introduction of a simple method for high-dynamic range photography [30]. In real outdoor and indoor environments, the range of intensities can be several orders of magnitude from direct sunlight, to the darkest regions in shadow. A conventional 8-bit image cannot capture this variation in dynamic range, and the resulting renderings will not be accurate. By using multiple exposures, Debevec and Malik [30] were able to obtain accurate high-dynamic range images of natural environments, that could then be used with global illumination in environment mapping [28]. Moreover, Debevec made available online a set of acquired environment maps [32], that have now been widely used throughout computer graphics. These advances made the technique practical throughout the gaming and movie industries.

Early work typically considered only mirror reflections from the surface (so the reflected direction can be directly used to index into the environment map). With the increasing popularity of environment maps, there was an increased need to support a variety of BRDF

models. Miller and Hoffman [93] had already discussed the use of preconvolved diffuse and specular reflection maps, and this was implemented in modern graphics hardware by Heidrich and Seidel [56]. In the same year, Cabral et al. [19] warped between multiple reflection maps, to implement more general BRDF models. A number of techniques for rapidly prefiltering and rendering with environment maps were subsequently introduced by Kautz and collaborators [70, 72]. A difficult problem in many of these prefiltering methods is the distortion introduced by any 2D parameterization of an environment map (since it really lies on the sphere).

## 3.2 Reflection as Convolution

In 2001, Ramamoorthi and Hanrahan [108, 110] and Basri and Jacobs [11] introduced a significant new theoretical result for environment maps. They showed that reflection from a curved surface, lit by an environment map, could be seen as a spherical convolution of the incident illumination and the reflective properties or BRDF of the surface. This is an exact result in the natural spherical domain, involving no distortions or parameterization issues. The convolution theory also implies a product formula in the frequency domain, in terms of spherical harmonic basis coefficients. These results formalize a signal-processing view of reflection, with the incident illumination signal being filtered by the BRDF kernel or filter, to obtain the reflected light. Moreover, if the BRDF is a broad lobe, such as in the case of Lambertian or diffuse reflection, the BRDF is a low-pass filter in the frequency domain. Therefore, only the lowest-frequency modes of the illumination are needed, enabling real-time computations.

These theoretical results for the first time formalized qualitative notions of reflection as convolution, that had been in the air for nearly two decades, going back to Miller and Hoffman [93]. They provided a unified framework for environment map rendering [108, 111], and a novel way to think about lighting. In this section, we will state the key results and intuition for diffuse or Lambertian surfaces lit by an environment map [11, 12, 108, 109]. Similar results can be derived for radially symmetric BRDFs like the Phong BRDF (symmetric about a central

direction like the reflection vector). Ramamoorthi and Hanrahan have also extended the convolution results to general BRDFs [110, 111], including anisotropic materials [112].

A simplified version of the reflection Equation (2.1) for Lambertian diffuse convex surfaces, lit by a distant environment map, is

$$B(\mathbf{n}) = \int_{\Omega_{4\pi}} L(\boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \mathbf{n}, 0)\, d\boldsymbol{\omega}_i. \tag{3.1}$$

In this expression, we have omitted the spatial dependence on $\mathbf{x}$, and the emission term. Note that the reflected light no longer depends directly on $\mathbf{x}$, nor on $\boldsymbol{\omega}_o$ (since we assume diffuse reflection, independent of outgoing angle). It only depends on the orientation or surface normal direction $\mathbf{n}$. The integral is now over the full sphere of directions for simplicity, with restriction to the upper hemisphere enforced by the max term in the integrand. The BRDF is simply a constant for Lambertian reflection, and this scale factor is omitted as it (and any spatial texture) can simply be used to multiply the final results. We have also dropped the subscript on the lighting environment map $L_i$ for simplicity. The kernel or filter in the above integral is the net transport function (BRDF times cosine term), that in this case is the *half-cosine*. This is the cosine of the angle between normal and incident directions in the visible hemisphere and zero when the light is below the horizon.

Intuitively, Equation (3.1) is analogous to a convolution, in which we center a kernel (the half-cosine $A(\mathbf{n},\omega) = \max(\mathbf{n} \cdot \omega, 0)$) about the surface normal, and integrate its product with the input signal $L(\omega)$. In fact, it can be shown [11, 109] that this is a spherical convolution of the incident lighting or environment map $L$, and the reflection kernel or *half-cosine* $A$. While the mathematics are somewhat more complicated, involving spherical harmonic basis functions [86], the techniques used are analogous to using Fourier analysis to understand the effects of a 1D convolution. In particular, there is a frequency–space convolution formula, where the spherical harmonic coefficients of the reflected light are simply a product of illumination and BRDF coefficients.

## 3.3 Spherical Harmonics

To derive and analyze the frequency–space convolution formula, we need to introduce some notation and the spherical harmonic basis functions. Spherical harmonics have also been widely used in PRT, which is why we devote this sub-section to describing them briefly. We note that they have been used previously in many applications of computer graphics rendering [18, 34, 117]. They are also closely related to the spherical polynomials used by Arvo [6] for his calculation of irradiance tensors — indeed, our expression for $B$ in Equation (3.1) corresponds directly to the irradiance. Spherical harmonics [34] and closely related polynomials [99] have also been used previously in some of the earliest work on precomputation-based relighting.

The spherical harmonics $Y_{lm}$ are functions of the elevation and azimuthal spherical coordinates $(\theta, \phi)$. $l \geq 0$ is the major index or frequency, and there are $2l + 1$ basis functions for each value of $l$, with $-l \leq m \leq l$. The $Y_{lm}$ are the analogue on the unit sphere to the Fourier basis functions on the line or circle. These basis functions arise in connection with many physical systems such as those found in quantum mechanics and electrodynamics. A summary of their properties can therefore be found in many standard physics textbooks [58, 60, 86]. The spherical harmonic $Y_{lm}$ is given by

$$\begin{aligned} Y_{lm}(\theta, \phi) &= N_{lm} P_{lm}(\cos\theta) \mathrm{e}^{Im\phi} \\ N_{lm} &= \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}, \end{aligned} \tag{3.2}$$

where $N_{lm}$ is a normalization factor. In the above equation, the azimuthal dependence is expanded in terms of Fourier basis functions (with $I = \sqrt{-1}$). The $\theta$ dependence is expanded in terms of the associated Legendre functions $P_{lm}$. The $Y_{lm}$ may be written either as trigonometric functions of the spherical coordinates $\theta$ and $\phi$ or as polynomials of the cartesian components $x$, $y$ and $z$, with $x^2 + y^2 + z^2 = 1$. In general, a spherical harmonic $Y_{lm}$ is a polynomial of degree $l$. Another useful relation is $Y_{l-m} = (-1)^m Y_{lm}^*$, where $Y_{lm}^*$ is the complex conjugate. The first three orders (we give only terms with $m \geq 0$) may be

written as follows ($I = \sqrt{-1}$),

$$\begin{aligned}
Y_{00} &= \sqrt{\frac{1}{4\pi}} \\
Y_{10} &= \sqrt{\frac{3}{4\pi}}\cos\theta &&= \sqrt{\frac{3}{4\pi}}z \\
Y_{11} &= -\sqrt{\frac{3}{8\pi}}\sin\theta e^{I\phi} &&= -\sqrt{\frac{3}{8\pi}}(x+Iy) \\
Y_{20} &= \frac{1}{2}\sqrt{\frac{5}{4\pi}}\left(3\cos^2\theta - 1\right) &&= \frac{1}{2}\sqrt{\frac{5}{4\pi}}\left(3z^2 - 1\right) \\
Y_{21} &= -\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta e^{I\phi} &&= -\sqrt{\frac{15}{8\pi}}z(x+Iy) \\
Y_{22} &= \frac{1}{2}\sqrt{\frac{15}{8\pi}}\sin^2\theta e^{2I\phi} &&= \frac{1}{2}\sqrt{\frac{15}{8\pi}}(x+Iy)^2 .
\end{aligned} \tag{3.3}$$

Figure 3.4 shows a visual representation of the first nine spherical harmonics, upto order 2. Note that the equations above represent

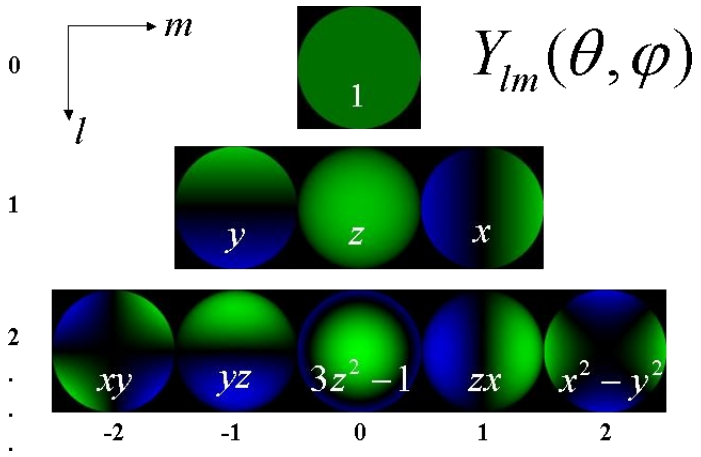


Fig. 3.4 Spherical Harmonics. We show the first three orders $l \leq 2$, or nine basis functions needed for irradiance or Lambertian reflection. In general, $l \geq 0$ and $-l \leq m \leq l$. Here, we see only the front of the sphere with green denoting positive values and blue denoting negative values. Spherical harmonics can be written either as trignometric functions of the spherical coordinates $(\theta, \phi)$ or as polynomials of the cartesian components $x$, $y$ and $z$. The first three orders therefore correspond simply to constant, linear and quadratic polynomials.

the standard complex form of the spherical harmonics, while numerical computations are often done with a real representation for simplicity. The real harmonics, shown in Figure 3.4, take the real and complex components respectively for $m > 0$ and $m < 0$, multiplying by $\sqrt{2}$ to keep the normalization. Thus, the real harmonics $Y_{11}$ and $Y_{1-1}$ are respectively $\left(\sqrt{3/[4\pi]}\right)x$ and $\left(\sqrt{3/[4\pi]}\right)y$ (with appropriate signs, based on the sign convention in effect). Terms with $m = 0$ are already real.

The spherical harmonics form an orthonormal basis. Therefore, any function over the unit sphere can be expanded in terms of the $Y_{lm}$. Expanding the lighting environment map $L$, the reflected light or irradiance $B$, and the transport half-cosine kernel $A(\theta) = \max(\cos\theta, 0)$,

$$\begin{aligned} L(\boldsymbol{\omega}_i) &= \sum_{l=0}^{\infty}\sum_{m=-l}^{l} L_{lm}Y_{lm}(\boldsymbol{\omega}_i) \\ B(\mathbf{n}) &= \sum_{l=0}^{\infty}\sum_{m=-l}^{l} B_{lm}Y_{lm}(\mathbf{n}) \\ A(\theta) = \max(\cos\theta, 0) &= \sqrt{\frac{2l+1}{4\pi}}\sum_{l=0}^{\infty} A_l Y_{l0}(\theta). \end{aligned} \tag{3.4}$$

In the last line, the normalizing factor is for simplicity in the final formula. Also note that since the kernel $A(\theta)$ is radially symmetric (no dependence on $\phi$), it is expanded only in terms of spherical harmonic basis functions $Y_{l0}$ with $m = 0$.

The convolution result allows us to write the spherical harmonic coefficients of the reflected light as a simple product of spherical harmonic coefficients of the incident illumination and the half-cosine,

$$\boxed{B_{lm} = A_l L_{lm}.} \tag{3.5}$$

Analytic formulae for the filter $A_l$ can also be derived [11, 109]. It can be shown that the $A_l$ decay rapidly, which is also shown graphically in Figure 3.5. In fact, it can be shown that this decay is as $l^{-5/2}$, with over 99% of the energy in the first three terms ($A_0$, $A_1$, and $A_2$). This corresponds to $l \leq 2$, for which the spherical harmonics are simply constant, linear, and quadratic polynomials of the cartesian

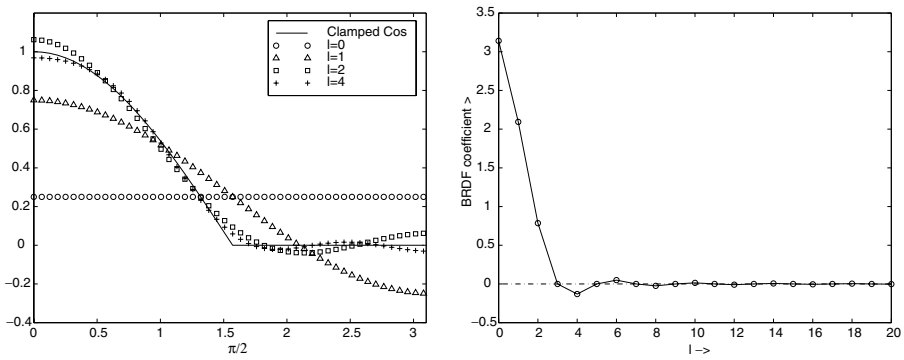


Fig. 3.5 Spherical harmonic representation of Lambertian filter. *Left:* Successive approximations to the half-cosine function for increasing spherical harmonic orders ($l = 0, 1, 2, 4$). For $l = 2$, we get a very good approximation. *Right:* The solid line is a plot of spherical harmonic coefficients $A_l$ for the half-cosine Lambertian reflection function. We see that odd terms vanish and even terms decay rapidly. Over 99% of the energy of the half-cosine function is captured by the first three orders $l = 0, 1, 2$ of spherical harmonics, corresponding to only nine terms.

coordinates. These represent a total of just nine spherical harmonic basis functions. *Irradiance or reflection from a diffuse surface is well approximated using only nine parameters, corresponding to the lowest frequency spherical harmonic basis functions.*

There have been a number of applications of this result to computer vision and computer graphics. Here, we focus only on environment map rendering. Given the nine parameter models, a very fast prefiltering algorithm can be developed to compute the nine spherical harmonic coefficients from an environment map. Thereafter, rendering just involves evaluating a quadratic polynomial for each pixel, which is trivial to implement in modern graphics hardware or software. This method, first introduced by Ramamoorthi and Hanrahan [108], has now become almost standard for video games and other interactive applications. This work has also been extended using general frequency domain convolution formulas for arbitrary BRDFs [111]. Example images produced by these algorithms are shown in Figure 3.6. Note that these techniques are not usually classified as precomputation methods, since minimal precomputation is required and the scene geometry can be changed dynamically.

Fig. 3.6 Spherical harmonic methods used for environment map rendering, based on the reflection as convolution framework. On the left, Lambertian and mirror reflections, along with standard diffuse textures, rendered using the nine term model for irradiance [108]. On the right, we show general materials (in this example, an anisotropic reflection model), using the convolution formula for general BRDFs [111]. These scenes can be rendered in real-time with environment maps. The techniques involved would not usually be classified as precomputation-based methods, since minimal precomputation is required and the scene geometry can be changed dynamically. However, these methods also do not include cast shadows or interreflections, as in modern PRT methods.

These results led to a great deal of interest in the community, in terms of including natural lighting in interactive computer-generated imagery. The earlier challenge of summing over all pixels of the environment map was addressed by showing that one really needed only the lowest frequency modes of the illumination. However, the convolution results assume convex surfaces, and do not include effects like cast shadows and interreflections. The precomputed radiance transfer method builds on spherical harmonic lighting approaches and the linearity of light transport by precomputing results (including shadows and interreflection) for each spherical harmonic basis function. Thereafter, rendering is possible with dynamic low-frequency illumination.

# 4

# Precomputed Radiance Transfer

Having developed the relevant background, we now describe the seminal 2002 paper by Sloan et al. [121], that introduced the term *precomputed radiance transfer*, and led to greatly increased interest in precomputation-based relighting methods. Historically, this is by no means the first paper on precomputation-based rendering, but builds strongly on previous work in image relighting [9, 35, 99, 132], as well as walkthroughs of diffuse scenes [34]. Even earlier, Airey et al. [2] used a linear combination of radiosity solutions computed with different sets of lights (but not a full environment as in the daylight models of [99]). More recently in 2000, Debevec et al. [29] had used a linear combination of directional lights, along with efficiency improvements based on compression, to relight images of faces. Precomputation had also been used to map light rays appropriately for highly specular reflections and refractions [49, 55].

However, there were two key innovations of Sloan et al. [121], that made it more powerful and applicable than previous methods, enabling widespread interest and adoption. The paper uses full environment maps for lighting instead of discrete light sources. The first innovation was to build on the spherical harmonic methods described in

the previous section, but precompute the shadowing and interreflection components, encapsulated in a *transfer function* on the object surface. This enabled "*real-time rendering in dynamic, low-frequency environments*" with shadows and interreflections. Second, the method worked directly on object geometry, computing the shading on vertices and reducing the calculation to a dot product that could be performed on the new generation of GPU graphics hardware. This enabled rapid adoption in video games, previewing and other interactive domains, achieving applicability well beyond static image relighting. Since this paper is the main precursor to much of the subsequent work on precomputation-based rendering, we will describe it and introduce the basic PRT concepts in some detail in this section.

## 4.1 Challenges and Assumptions

Before delving into the details of the method, it is important to understand both the challenges of real-time high quality rendering, and the assumptions and limitations of the solution proposed.

As noted in [121], real-time realistic rendering encounters three main difficulties: *BRDF complexity* or modeling the complex reflectance properties of real materials, *light integration complexity*, involving the integration of illumination over the entire visible hemisphere, and *light transport complexity* to account for shadows, interreflections and other global lighting effects. There is a significant body of work on complex analytic and measured BRDF models. Standard offline renderers address the problem of light integration complexity by breaking the illumination environment into a set of point sources or directions, adding up the contributions from these directions. However, light integration remains a major bottleneck for real-time applications that cannot afford the expensive summation over light source directions, especially for area sources or environment maps. Indeed, a 2D image representation of an environment map contains thousands to millions of pixels.

The spherical harmonic convolution framework discussed in the previous section nicely addresses some of these challenges. Low-frequency lighting environments (and BRDFs) can be represented using only a few spherical harmonic basis functions, making light integration or

summation practical — in fact, Sloan et al. [121] show how this reduces to a simple vector dot product that can be evaluated in graphics hardware. Moreover, spherical harmonics are well adapted to the sphere, and can be dynamically rotated without aliasing (a spherical harmonic $Y_{lm}$ on rotation can be written as a linear combination of spherical harmonics $Y_{lm'}$ with the same major index $l$). Much the same mathematics as the Lambertian case, but using more spherical harmonic terms, can also be applied to glossy materials enabling specular reflections [110, 111]. This suggests that spherical harmonic lighting respresentations have much potential for real-time rendering. However, the convolution framework is limited to convex objects without interreflections or shadowing.

The work of Sloan et al. [121], thus focuses on increasing *light transport complexity* in real-time rendering. They precompute a *transfer function* at each vertex of the object, that encodes how the object responds to a particular spherical harmonic lighting basis function, taking shadows and other effects into account. For convex objects, the transfer function is simply a cosine-weighted integral over the BRDF (the *half-cosine* function for Lambertian objects). However, for concave objects, the transfer function can include visibility and the effects of self-shadowing, as well as interreflections. While computing this transfer or light transport function is expensive, it is a preprocess that can be done with standard offline renderers. The resulting transfer functions are then stored densely on the object surface, usually at its vertices, as vectors or matrices corresponding to the response to each spherical harmonic basis illumination. The linearity of light transport holds in any basis, including spherical harmonics, and light integration reduces at run-time to a simple dot-product (typically involving 9–25 elements) of transfer coefficients and those of the incident illumination or environment map. The basic idea is shown pictorially in Figure 4.1.

A number of assumptions must be made for real-time performance. First is the key assumption of *low-frequency* lighting environments. Because the final image is a convolution of lighting and BRDF for convex objects, high frequencies in the BRDF may also be lost if the illumination is limited to low-order spherical harmonics. Moreover, the method is also often limited to very smooth soft shadow effects.

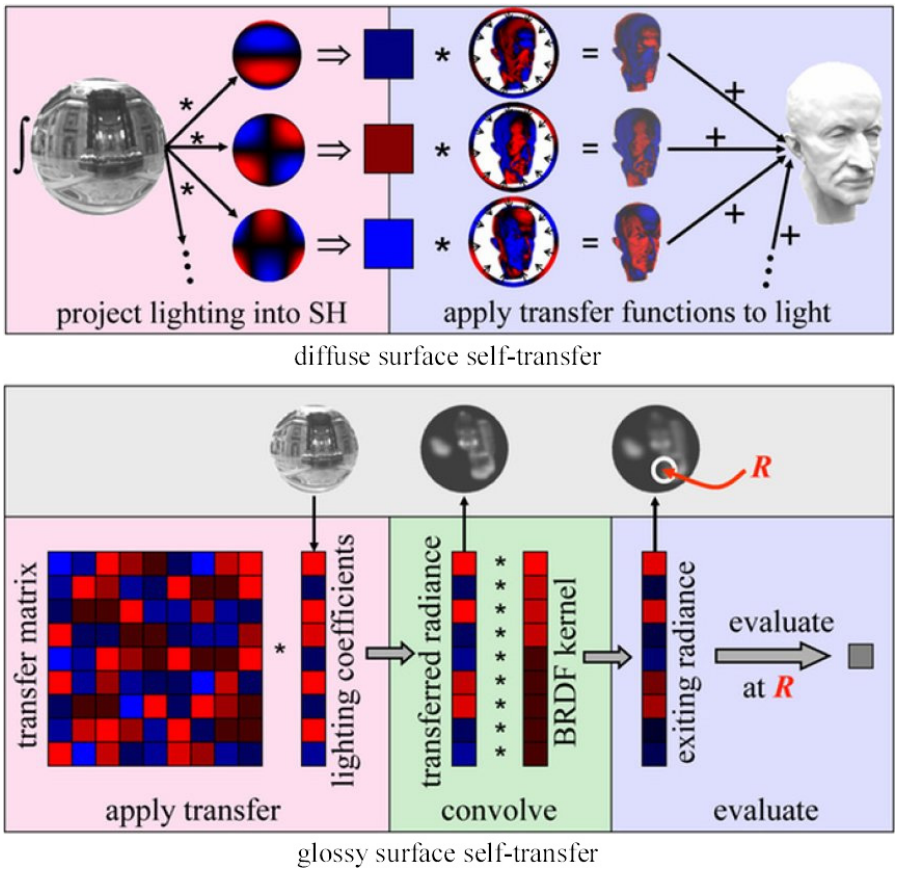


Fig. 4.1 Steps of PRT algorithm (Fig. 2 in [121]). Red denotes positive spherical harmonic coefficients while blue is negative. *Top*: Diffuse radiance transfer. The lighting is projected into spherical harmonic coefficients, which scale the precomputed transfer functions over the object surface. The final result is obtained by summing the scaled transfer functions, or taking a dot-product between vectors representing lighting and transfer coefficients at each pixel. *Bottom*: Glossy materials. In this case, the visibility of the surface leads to a transfer matrix at each spatial location that transforms the environment map to local incident radiance. This is then convolved with the BRDF kernel to obtain an outgoing radiance distribution, that is then evaluated at the reflection vector.

In practice, any environment map can be used, rotated relative to the object, or blended, but the results will be accurate only for the low-frequency components of the lighting. This assumption enables a small number of spherical harmonic terms to be used, and the authors typically go up to $l^* = 4$ or 25 spherical harmonics (there are a total of

$(l^* + 1)^2$ spherical harmonics for order $l \leq l^*$). While this method captures more complex effects than the nine spherical harmonics ($l^* = 2$) used for Lambertian reflection, the results remain low-frequency with soft shadows and broad specular reflections.

A second assumption is that the geometry of the objects remains static, so that a precomputation can be performed on the surface. In this paper [121], glossy reflections are allowed, and the object material properties can also be changed dynamically (assuming a low-frequency Phong-like BRDF). However, lighting and view cannot simultaneously be varied in real-time — subsequent improvements to relax this assumption [120] require the BRDF to be fixed, as does correct computation of interreflections.

## 4.2 Precomputed Radiance Transfer Algorithm

Consider the reflection Equation (2.1), setting the emission term to 0. Let us further assume lighting from a distant environment map. The incident illumination $L_i(\mathbf{x}, \boldsymbol{\omega}_i)$ at a point $\mathbf{x}$ is then simply $L(\boldsymbol{\omega}_i)V(\mathbf{x}, \boldsymbol{\omega}_i)$ where $L$ is the environment map (only a function of direction), and $V(\mathbf{x}, \boldsymbol{\omega}_i)$ is a binary visibility function that encodes if direction $\boldsymbol{\omega}_i$ is visible from spatial location $\mathbf{x}$. The reflection equation then becomes,

$$\boxed{B(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega_{4\pi}} L(\boldsymbol{\omega}_i)V(\mathbf{x}, \boldsymbol{\omega}_i) \left[\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}), 0)\right] d\boldsymbol{\omega}_i,} \tag{4.1}$$

where we have also dropped the spatial dependence in the BRDF $\rho$. This simplified form of the reflection equation is widely studied and used in PRT. The integrand now involves a product of three terms, the lighting environment map, the visibility and the BRDF of the surface, integrated over all incident directions.

### 4.2.1 Lambertian Objects

We start by considering diffuse objects. In this case, the BRDF is just a constant (given by $1/\pi$ times the albedo), and we now obtain

$$B(\mathbf{x}) = \int_{\Omega_{4\pi}} L(\boldsymbol{\omega}_i)V(\mathbf{x}, \boldsymbol{\omega}_i)\rho \max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}), 0)\, d\boldsymbol{\omega}_i. \tag{4.2}$$

Note that if the visibility term is simply 1 (fully visible), we essentially obtain Equation (3.1) for diffuse convex surfaces (if we express $B$ in the above equation in terms of the normal $\mathbf{n}$, rather than $\mathbf{x}$). In that case, the equation above can be simplified to the spherical harmonic convolution formula in Equation (3.5). However, the goal of PRT is to go beyond this result and also include the effects of cast shadows.

Now, let the lighting be expressed in spherical harmonics, so that

$$L(\boldsymbol{\omega}_i) = \sum_{j=1}^{(l^*+1)^2} \alpha_j Y_j(\boldsymbol{\omega}_i), \tag{4.3}$$

where we use a single index $j = (l, m)$ to subscript the spherical harmonic for simplicity. Going up to some major index $l^*$, we have a total of $(l^* + 1)^2$ spherical harmonic terms. Equation (4.3) simply represents the lighting as a linear combination of spherical harmonic basis functions, corresponding closely to the generic form introduced in Equation (2.6). Equation (4.2) can then be written as,

$$B(\mathbf{x}) = \sum_j \alpha_j \int_{\Omega_{4\pi}} Y_j(\boldsymbol{\omega}_i) V(\mathbf{x}, \boldsymbol{\omega}_i) \rho \max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}), 0)\, d\boldsymbol{\omega}_i, \tag{4.4}$$

where we have exchanged the order of summation and integration. Note that $\rho$ is just a scalar in this case, and we could also move it out of the integral. However, we choose not to do so, since this will no longer be possible for interreflections and more complicated BRDFs. Finally, note that the above derivation and expressions hold for any orthogonal basis and we could replace $Y_j$ with Fourier series, wavelets and other forms, as in some subsequent work in PRT.

In the above equation, the integrand does not depend on the lighting (or $\alpha_j$) and can therefore be precomputed, defining a transfer function

$$T_j(\mathbf{x}) = \int_{\Omega_{4\pi}} Y_j(\boldsymbol{\omega}_i) V(\mathbf{x}, \boldsymbol{\omega}_i) \rho \max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}), 0)\, d\boldsymbol{\omega}_i. \tag{4.5}$$

Thereafter, rendering reduces to evaluating

$$B(\mathbf{x}) = \sum_j \alpha_j T_j(\mathbf{x}). \tag{4.6}$$

In practical computations, there are several important design choices and algorithmic steps that these relations lead to naturally. First, we must precompute the transport coefficients $T_j(\mathbf{x})$. We refer readers to the paper by Sloan et al. [121] for details, noting that Equation (4.5) is an integral that can be evaluated by standard Monte Carlo techniques. In the next section, we will briefly describe ray-tracing and rasterization methods to evaluate high-frequency visibility in the context of all-frequency relighting [96]. Next, we must store the $T_j(\mathbf{x})$; the transport coefficients are stored densely over an object's surface, usually at the vertices of a finely tesselated mesh. Note that this does involve significant storage, and a number of subsequent techniques develop compression schemes — one of the earliest examples is by Lehtinen and Kautz [80]. Finally, the rendering in Equation (4.6) is essentially a dot-product at each spatial location $\mathbf{x}$, given by $\sum_j \alpha_j T_j = \boldsymbol{\alpha} \cdot \mathbf{T}$, where $\boldsymbol{\alpha}$ and $\mathbf{T}$ are respectively, a vector of spherical harmonic coefficients for illumination and transport functions. This dot-product involves 9 to 25 terms for low-frequency lighting ($l^* = 2$ or $l^* = 4$) and can be evaluated for each vertex or pixel in real-time using a GPU shader.

So far, we have only considered shadowed irradiance, without interreflections. When diffuse interreflections are considered, we must consider the rendering rather than reflection equation. However, by linearity of light transport, $T_j(\mathbf{x})$ can still be interpreted as the appearance of surface point $\mathbf{x}$ when lit by spherical harmonic illumination $Y_j(\boldsymbol{\omega}_i)$. As such, it can be precomputed by any global illumination renderer, after which rendering can still be done using Equation (4.6). Examples of diffuse shadowed and interreflected renderings can be seen in Figure 4.2.

### 4.2.2 Specular Materials

For glossy objects, the reflected radiance $B(\mathbf{x}, \boldsymbol{\omega}_o)$ will be view-dependent. Here, we describe the approach of [121], that addresses low-frequency Phong-like BRDFs, symmetric about a central direction such as the reflection vector. An early extension to general BRDFs is presented by Kautz et al. [71].


Fig. 4.2 Results of diffuse (left) and glossy (right) radiance transfer. A comparison of images from Sloan et al. [121] without shadows (or interreflections), and rendered using PRT with shadows and interreflections. The PRT algorithm enables global illumination effects like soft shadows and interreflections in real-time rendering, with dynamic low-frequency lighting.

The basic approach follows certain steps, analogous to the **G** and **K** operators we described for light transport in section 2.2. These steps are shown diagrammatically in the bottom row of Figure 4.1. First, we must transport the incident environment map to the local incident radiance at the surface. For simple reflections, this just corresponds in the angular domain to the product of lighting and visibility, $L(\boldsymbol{\omega}_i)V(\mathbf{x},\boldsymbol{\omega}_i)$.

The challenge is computing this product in the spherical harmonic domain, where it is no longer a simple point-wise multiplication. In fact, projecting a product of two functions, each described in spherical harmonics, involves a quantity known as the triple-product integral [97]. A simpler approach is to appeal once again to the linearity of light transport. If the scene is lit by a single spherical harmonic function, this will lead to some incident radiance function after multiplying by visibility. We can simply precompute and sum these results for general lighting. In other words, a spherical harmonic $Y_k(\boldsymbol{\omega}_i)$ for the incident

lighting gives rise to a light distribution at the local surface of

$$Y_k(\boldsymbol{\omega}_i) \rightarrow L_i(\mathbf{x}, \boldsymbol{\omega}_i) = \sum_j M_{jk}(\mathbf{x}) Y_j(\boldsymbol{\omega}_i), \tag{4.7}$$

where $Y_k$ is the global spherical harmonic lighting, that leads to a lighting at the local surface $L_i(\mathbf{x}, \boldsymbol{\omega}_i)$ in terms of spherical harmonics $Y_j(\boldsymbol{\omega}_i)$. $M_{jk}(\mathbf{x})$ controls how much of the energy of $Y_k$ gets distributed to spherical harmonics $Y_j$ for that spatial location $\mathbf{x}$. Since $M$ can be thought of as a matrix at each point, this approach is also known as representing visibility using a *transfer matrix* in spherical harmonics. Finally, we can sum over the incident lighting distribution,

$$L_i(\mathbf{x}, \boldsymbol{\omega}_i) = \sum_j \beta_j(\mathbf{x}) Y_j(\boldsymbol{\omega}_i) \quad \beta_j(\mathbf{x}) = \sum_k M_{jk}(\mathbf{x}) \alpha_k, \tag{4.8}$$

where the $\alpha_k$ are the spherical harmonic coefficients of the incident lighting. The equation for $\beta_j$ above is effectively a matrix-vector multiplication of $\mathbf{M}$ (corresponding to visibility and/or interreflections) and the environment map lighting coefficients $\alpha_k$.

Once we have the incident light coefficients $\beta_j$, they can be attenuated or filtered based on the BRDF (such as the Phong exponent). In the spherical harmonic domain, this is a diagonal matrix, with each major index $l$ attenuated by the same amount, and can be represented by BRDF coefficients $P_j$. The specific values can be computed analytically for Phong and Torrance-Sparrow BRDFs [110]. The result of convolution with the BRDF gives the reflected light. Finally,

$$B(\mathbf{x}, \hat{\boldsymbol{\omega}}_o) = \sum_j P_j \beta_j(\mathbf{x}) Y_j(\hat{\boldsymbol{\omega}}_o), \tag{4.9}$$

where we use $\hat{\boldsymbol{\omega}}_o$ to note that the sphere of reflected light is indexed into based not on the outgoing direction *per se*, but on a suitable direction such as the reflection of the view about the surface normal. Some techniques explicitly include spherical harmonic rotation matrices to move between local and global coordinate frames.

The combined light and view-dependence of the result can be written in matrix form, if we define a column vector $\boldsymbol{\gamma}$ with $\gamma_j = Y_j(\hat{\boldsymbol{\omega}}_o)$. In this case,

$$B(\mathbf{x}) = \boldsymbol{\gamma}^t \cdot \mathbf{P} \cdot \mathbf{M}(\mathbf{x}) \cdot \boldsymbol{\alpha}, \tag{4.10}$$

where $\boldsymbol{\gamma}$ is related to the viewing direction, and $\boldsymbol{\alpha}$ to the illumination. Note that this matrix equation must be applied at each spatial location $\mathbf{x}$. Also note that if we fix one of the light or view, it can be pre-multiplied to collapse the net transport to a vector (e.g., define $\mathbf{T}(\mathbf{x}) = \boldsymbol{\gamma}^t \cdot \mathbf{P} \cdot \mathbf{M}(\mathbf{x})$), with rendering reduced to a dot product as in Equation (4.6). In fact, Sloan et al. [121] are unable to perform the matrix multiplication in Equation (4.10) directly on the GPU, and can achieve real-time performance only by fixing either lighting or view. Examples of specular radiance transfer from Sloan et al. [121] are shown in Figure 4.2.

### 4.2.3 Discussion

In terms of performance, the PRT method makes a number of careful design tradeoffs. The offline precomputation time can be significant, usually 1–2 hours for the 50 K–150 K vertex models used in [121]. Thereafter, rendering speed is fast and approaches 100 frames per second, for true real-time performance. Storage is a concern, and can occupy megabytes (and gigabytes when extended to all-frequency effects [96]). However, for low-order spherical harmonics, the data is compact enough to be stored on modern graphics cards, at least for diffuse transfer. Methods based on this technique are already in production use today in commercial game studios.

Besides the basic method presented so far, there are a number of extensions developed in [121]. Instead of assuming the lighting is distant, it can be sampled at a number of points on the surface to allow for near-field illumination. Graphics hardware can be used both for this sampling, as well as in actual rendering. There is also an example demonstrated of self-transfer for volumetric models, as well as transfer within a neighborhood, allowing an object to cast soft shadows onto a terrain underneath as it moves. The method can even be used for a ring with low-frequency caustics. In all, the PRT paper presented a new capability in real-time rendering, that led to a great deal of excitement as new visual effects like environment lighting, soft shadows, and diffuse interreflections entered the real-time domain.

The PRT method as presented does have a number of limitations, that subsequent work seeks to address, and which the ensuing sections survey. There is a fundamental limitation to low-frequency effects. By using other bases than spherical harmonics (such as wavelets and radial basis functions), higher-frequency effects can be achieved. Compression is a critical issue, especially when storing transport matrices over a mesh. New techniques such as clustered principal components [120] for spatial compression were invented for this purpose. In terms of functionality, one goal is simultaneous light and view motion. The others are the ability to consider dynamic scenes, and ultimately to change any of the lighting, view, material and geometry. The remaining sections of this survey detail many of these extensions that have greatly broadened the applicability of the precomputed radiance transfer framework.

# 5

# Basis Representations and Compression

The precomputed radiance transfer method described in the previous section took a groundbreaking step towards the goal of realistic rendering in real-time. Two key limitations are to low-frequency effects only, and the size of the precomputed datasets, especially if we seek to extend the method to higher resolutions and frequencies. In this section, we survey a significant body of work that seeks to address these limitations. We start by giving a conceptual overview, motivated by the development of all-frequency effects and wavelet representations [96]. We then discuss a variety of compression schemes that extend the applicability of PRT, and simultaneously reduce run-time costs and data storage.

## 5.1 Overview: Relighting as Matrix-Vector Multiplication

Following Ng et al. [96] and earlier work on image relighting, we define the relighting problem conceptually by starting with the definition in Equation (4.1). First, assume fixed view or image relighting, so the outgoing direction $\boldsymbol{\omega}_o(\mathbf{x})$ is fixed, and define a light transport operator

$$T(\mathbf{x},\boldsymbol{\omega}) = V(\mathbf{x},\boldsymbol{\omega})\rho(\boldsymbol{\omega},\boldsymbol{\omega}_o(\mathbf{x}))\max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}),0). \tag{5.1}$$

In this case, the integral for $B$ in Equation (4.1) is given simply by the product of the incident illumination $L$ and light transport $T$. If we discretize, this can be written as a matrix equation,

$$B(\mathbf{x}) = \sum_i T(\mathbf{x}, \boldsymbol{\omega}_i) L(\boldsymbol{\omega}_i), \tag{5.2}$$

which can be compactly represented as

$$B = \mathbf{T}L, \tag{5.3}$$

where $B$ and $L$ are column vectors for the computed radiance (or image) and incident illumination (or environment map), respectively, while $\mathbf{T}$ is a matrix. Each row of $\mathbf{T}$ corresponds to one image pixel or geometric vertex, and each column to a particular lighting direction. For high fidelity effects, the number of rows may be approximately 300,000, either for many vertices in complex geometry or for a $512 \times 512$ image. The number of columns may be approximately 25,000 corresponding to a $6 \times 64 \times 64$ cubemap representation of the lighting environment.

As noted by Ng et al. [96], the matrix $\mathbf{T}$ can be precomputed for a static scene, either by raytracing images from individual lights/pixels in the environment map (corresponding to columns of $\mathbf{T}$) or by rasterizing the visibility for direct lighting for each pixel in the image or vertex on geometry (corresponding to rows of $\mathbf{T}$). The rasterization method can be sped up using modern GPUs and an optimized technique can easily reduce precomputation time to under half an hour in many cases. However, it is limited to direct lighting. The ray-tracing approach is usually slower, but can incorporate global illumination for a variety of visual effects including caustics.

In the ideal case, we would like to be able to work directly with Equation (5.3). Note, however, that the matrix $\mathbf{T}$ has dimensions $300{,}000 \times 25{,}000$ or nearly $10^{10}$ elements, which is also the cost of the matrix-vector multiplication. This is usually impractical for real-time performance, and requires significant compression even for storage.

Existing methods for image relighting and PRT techniques can be seen as various compression and approximation schemes to deal with the challenge of complexity. First, note that Equation (5.3) holds in any basis representation, not just the canonical situation where

columns represent pixels in the environment map, and the lighting is stored directly in a pixel representation. The PRT algorithm of Sloan et al. [121] effectively computes Equation (5.3) in a spherical harmonic basis. Furthermore, they use only the first few columns of $\mathbf{T}$ in that basis (typically 9–25 columns) and the corresponding elements of the lighting $L$. This can be seen as employing a *linear approximation* [89], where the terms to be used (low-frequency spherical harmonic coefficients) are chosen in advance.

## 5.2 All-Frequency Relighting: Wavelet Lighting Approx.

The challenge of all-frequency relighting, as originally proposed by Ng et al. [96] in 2003, is to go beyond the low-frequency spherical harmonics and be able to render shadows and other effects at both high and low frequencies. The key idea is to approximate the lighting dynamically at run-time in a wavelet representation [89]. Wavelets are especially suited as they are a multiresolution transform, at a variety of different scales. Thus, small wavelets can be used for compact sources like the sun or desk lamps, and large wavelets for broad area distributions like skylight. Wavelets were introduced into computer graphics in the mid-1990s [124] and have since been widely used in many applications.

More formally, the simplest basic or mother wavelet is the Haar wavelet in 1D,

$$\begin{aligned} \psi(u) = \quad & +1, \quad 0 \le u < 1/2, \\ & -1, \quad 1/2 \le u < 1, \\ & 0 \quad \text{otherwise.} \end{aligned} \tag{5.4}$$

For both its simplicity and its compact support, Haar has frequently been used in PRT. However, it is also possible to use smoother wavelets with larger supports, such as Daubechies wavelets [27].

To create wavelets across a variety of scales, we apply translation and dilation to the mother wavelet,

$$\psi_{jk}(u) = 2^{j/2}\psi(2^j u - k), \tag{5.5}$$

where $j$ is the level of the wavelet (starting with 0 for the coarsest level and increasing for finer levels to $j = J$), and $k$ is the translation

with $0 \leq k < 2^j$. The basis is completed with a scale function, which for Haar is $\phi(u) = 1$ when $0 \leq u < 1$.

The wavelets form a basis into which any signal can be projected, just like Fourier series or spherical harmonics, i.e., any signal can be written as:

$$f(u) = \alpha_0 \phi(u) + \sum_{j,k} \alpha_{jk} \psi_{jk}(u), \tag{5.6}$$

where $\alpha_{jk}$ are the wavelet coefficients, and the first term of the above equation is the scale function $\phi(u)$. In fact, a wavelet transform is very efficient and can be done in linear time. It is a multi-level transform with $J$ steps. Each step involves finding scale and different coefficients for that level. For Haar, we simply compute the average of two neighboring values and their difference. The averages are then transformed further in the next step. A more complete introduction to wavelets is given in a number of standard texts [33, 89, 124].

The wavelets can also be extended to 2D, using either a standard or non-standard wavelet decomposition. In the standard approach, we first transform all rows of an image, then all columns. In the non-standard method, that usually provides better compression, row and column transformations are interleaved. Of course, illumination technically resides on a sphere, and a framework for spherical wavelets [116], as well as extensions of Haar wavelets to the sphere [82], have been proposed. However, for simplicity, most work in PRT has actually used Haar wavelets on a planar cubemap representation of the sphere.

In terms of approximation, we pick the most important wavelets for a given illumination. Since these are not determined *a-priori* or restricted to low-frequency terms, it is known as *non-linear wavelet approximation* [33], and captures all-frequency effects. Note that the reconstructed lighting is still linear in the wavelet coefficients. The term "non-linear approximation" can be confusing, and relates to the fact that the wavelet approximation for a sum of two signals may not be the same as the sum of the individual wavelet approximations (because different wavelet coefficients may be chosen in the two cases). In simpler terms, which wavelet coefficients are to be chosen (high-frequency, low-frequency and where) is selected only at run-time. A variety of metrics

to choose the most important wavelets can be applied, including simply the $L_2$ norm, area-weighted selection, and transport-weighted selection [96]. Selecting coefficients based on area-weighting by the support of the wavelet is found to perform best, which prioritizes low-frequency lights that are the most important for largely diffuse scenes.

At its core, this is a very simple idea, to speed up the relighting computation by approximating the lighting $L$. As shown in Figure 5.1, a key observation is that excellent results can be obtained, using only 1% of the wavelets, or about 100–200 basis coefficients, which is one to two orders of magnitude less than the number of spherical harmonic terms required to produce comparable results. (Even further compression to about 30–100 wavelet coefficients is possible by exploiting the temporal coherence of lighting from frame to frame [102].) Of course, these are results for typical scenes, and the exact compression is data-dependent, and varies with the BRDFs used. Moreover, the convergence in wavelets is exponentially faster than in spherical harmonics. Note also that the lighting itself need not be reconstructed with very high accuracy, since it will effectively be low-pass filtered by the light

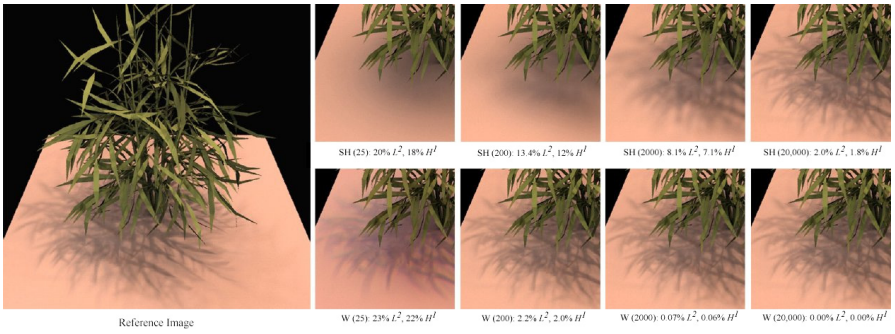


Fig. 5.1 Image from all-frequency relighting of a plant scene [96]. The illumination environment map is in St. Peter's Basilica and displays intricate sharp shadows from small windows. The reference image is on the left, and we compare shadow fidelity using a spherical harmonic approximation (top row) and wavelet approximation (bottom row). Error is reported using both the conventional $L^2$ norm, and the $H^1$ norm that takes gradients into account, as useful for detailed shadow regions. The number of terms used is given in parentheses below each picture. Note that for high quality shadows, we need about 200 wavelet coefficients (W) for the lighting, but two orders of magnitude more (20,000) terms in spherical harmonics (SH). The plant model is courtesy of O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr, and P. Prusinkiewicz.

transport operator — in fact, in many cases only 50% accuracy for the illumination gives final images very close to ground truth.

The wavelet approach also has many relations with the use of sparsity to accelerate matrix-vector computations, since relighting can be reduced to a matrix-vector multiplication as per Equation (5.3). There is a key difference however. In sparse matrix calculations, the sparsity of the matrix is the key factor. In contrast, we seek to *sparsify the vector* rather than the transport matrix $\mathbf{T}$. In practice, the matrix is quantized and zeros discarded, but this is a relatively minor effect (keeping 20%−30% of the entries), compared to the 0.1%−1% of the terms stored in the vector. (It should be noted that later work [143] has indicated that it may be possible to symmetrically apply an aggressive non-linear wavelet compression to matrix rows, while keeping all the terms of the lighting.)

The initial work of Ng et al. [96] used Haar wavelets for their simplicity and ease of basis projection. A variety of more complex orthogonal wavelets such as Daubechies filters [27] can also be used, as in [14]. Note that wavelets are limited in being aligned to the cubemap axes. Therefore, they are inefficient in capturing features not aligned with the cubemap grid. Alternative non-linear representations like spherical radial basis functions or SRBFs [133] can provide better performance for all-frequency relighting in these circumstances, as can kernel-based schemes [44]. In fact, SRBF representations have also been shown to provide compact descriptions of normal map distributions for filtering [51]. As an aside, we note that improved basis representations are not limited to wavelets. Several extensions of spherical harmonics to the hemisphere, including Zernike polynomials [74] and hemispherical harmonics [37] have been proposed.

The all-frequency relighting technique was able to produce some of the highest fidelity images to be rendered in real-time (see Figure 5.1 and Figure 1.1 right). The main drawback of this method is the size of the precomputed matrices (up to 1 GB for intricate shadowing), and the relatively bulky matrix-vector multiplication that is difficult to implement on the GPU (however, later work [52, 145] has made progress on hardware relighting approaches). For this reason, the technique has not yet been included in commercial software. The initial approach

was limited to image relighting and diffuse geometry relighting. In Section 6.1.1, we discuss an extension using wavelet triple products to enable all-frequency relighting of glossy objects with changing view [97].

## 5.3 Transport Compression: Clustered Principal Components

So far, wavelet compression has been applied only to the lighting, and consequently to the columns of the light transport matrix $\mathbf{T}$. However, several pixels in the image (or vertices of object geometry) will have similar transport functions (rows of the matrix $\mathbf{T}$), and we seek to exploit this spatial or signal coherence. One obvious method is to also wavelet transform in the image domain (rows of $\mathbf{T}$). As noted in [21], this must be done carefully to actually obtain compression, and the gains are modest. An alternative widely used method is to break the image or geometric model into clusters or blocks, finding a local subspace approximation within each cluster. The technique, known as Clustered Principal Component Analysis or CPCA was originally introduced in 2003 by Sloan et al. [120] for low-frequency spherical harmonic PRT with both light and view-dependence. For conceptual simplicity, we first explain it in the context of the block-PCA method (including high frequencies) described by Nayar et al. [95].

### 5.3.1 Block PCA for Image Relighting

As far back as the initial seminal work on image relighting in the mid-1990s, it was recognized that compression was an issue and Singular Value Decomposition (SVD, used here interchangeably with Principal Component Analysis or PCA) was used by Dorsey et al. [35]. However, for complex all-frequency effects, $\mathbf{T}$ itself does not have low rank. Therefore, hundreds to thousands of principal components may be required. The key insight in [95, 120] is that the transport matrix is *locally low-rank*, i.e., a particular cluster or local image patch or block has a much lower-dimensional representation than the full matrix.

Formally, assume the image is divided into patches $j$, with each patch containing $p_j$ pixels. Let $T_j$ denote the transport matrix for that patch alone, having dimensions $p_j \times l$, where $l$ is the lighting resolution.

An SVD decomposition of $T_j$ approximated with $n_j \ll l$ eigenmodes is

$$T_j \approx U_j S_j V_j^t, \tag{5.7}$$

where $U_j$ is a $p_j \times n_j$ matrix, $S_j$ is a $n_j \times n_j$ diagonal matrix of eigenvalues and $V_j$ is a $l \times n_j$ matrix, as is standard for SVD. It can be shown that significant compression can be achieved, with $n_j \ll l$ for small patches. For example, for a patch of $16 \times 16$ pixels so that $p_j = 256$, using $n_j = 10$ basis functions often suffices [95].

We need to consider the computational savings of this transformation. First, if no compression were performed, the matrix-vector multiplication has cost proportional to the dimensions of $T$ or $p \times l$ for the full image — or more useful, a cost of $l$ per pixel.

The reduced-dimension approximation for $T_j$ consists of two steps. First, $V_j^t$ transforms $l$ lighting directions to $n_j$ local basis coefficients. Indeed, this is the key idea of this work — for each cluster or image patch, there is a local lighting basis where a low term approximation suffices. For the diffuse Lambertian unshadowed case, a global low-frequency spherical harmonic basis suffices, but a low-dimensional global basis will not be adequate in the general case. However for a local patch, an appropriately chosen local basis suffices. This operation $S_j V_j^t L$ needs to be done only once for the entire patch, and has total cost $n_j l$.

Then for each pixel, these local coefficients are appropriately weighted with per-pixel weights to get the pixel intensities by multiplication with $U_j$, that has a net cost $p_j n_j$. The total cost $c_j$ for the patch is therefore,

$$c_j = n_j l + n_j p_j = n_j (l + p_j). \tag{5.8}$$

The cost per-pixel can be obtained by dividing by $p_j$ and is given by

$$\frac{c_j}{p_j} = n_j \left(1 + \frac{l}{p_j}\right). \tag{5.9}$$

This expression should be contrasted to a per-pixel cost of $l$ if using the lighting directly without approximating the transport matrix.

The savings can be of the order of $n_j/l$, which is the compression achieved by the locally low-rank approximation within each cluster or

local image block. Note, however, the second term $n_j l/p_j$ for the overhead involved in projecting onto the local basis. This overhead can be significant, since high-resolution lighting can easily involve $l = 25{,}000$ directions. Therefore, Nayar et al. [95] do a second SVD by stacking the local basis functions $V_j$, giving rise to many fewer basis functions $l' \ll l$. Of course, this now requires a global conversion between the pixel basis and the new global basis, with cost $l' \times l$. However, this need only be done once per frame, and can in fact be sped up using a sparse wavelet representation for $l$ (thereby combining wavelet lighting compression and local PCA compression of the transport).

As shown in Figure 5.2 left, the total cost is the sum of the per-pixel multiplication and per-patch projection (overhead) costs. Therefore, there is an interesting question of how big to make the patch. If we consider patches the size of an entire image, we will not obtain much compression and the per-pixel cost $n_j$ will be high. But the overhead to project onto the local patch is amortized by all the pixels in the image (division by $p_j$) and is low. At the other extreme, if the patch consists of a single pixel, only one basis function $n_j = 1$ will be needed. But the overhead will be large, on the order of $l$, leading to no increase in efficiency. Early work determined the appropriate intermediate patch size largely in an *ad-hoc* or trial and error fashion. More

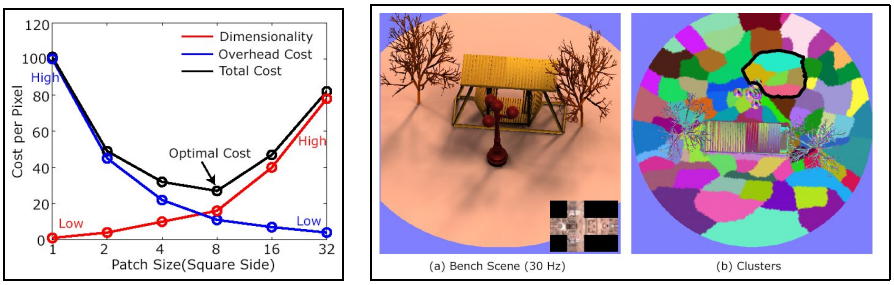


Fig. 5.2 *Left*: Conceptual representation of how rendering costs vary with patch size. As the patch size increases light transport dimensionality or per-pixel cost $n_j$ increases (red), while the per-patch overhead (blue) decreases. The overall cost (black) is minimum at an intermediate point where the increase in dimensionality balances the decrease in the per-patch overhead. *Right*: Results, with changing lighting and view, rendered at real-time frame rates with CPCA. We show both the scene, and the resulting clusters used. The figure is from Mahajan et al. [87], and the bench model is courtesy of R. Overbeck.

recently, Mahajan et al. [87] have conducted a complete thoretical analysis of locally low dimensional light transport, including a number of practical methods to determine the optimal patch or cluster size.

### 5.3.2 CPCA for Spherical Harmonic PRT

The basic approach of Nayar et al. [95] can be denoted as block-PCA, because its patches or clusters are small $8 \times 8$ or $16 \times 16$ blocks of the image. The more general work of Sloan et al. [120] develops a CPCA method, where the clusters can be arbitrary regions (in their case clusters of vertices on object geometry), and are determined adaptively to best reduce the error. This seminal work [120] is motivated by the need for real-time variation of both illumination and viewpoint, which was not possible in the original PRT paper [121].

The light transport matrix is now stored in spherical harmonics rather than in the pixels of the environment map. Each row (corresponding to a vertex on object geometry) has 625 columns ($25 \times 25$ spherical harmonics for light and view dependence). Apart from this, the conceptual framework is very similar to Equation (5.3). Note, however, that the transport matrix now also encodes view-dependence. Therefore, similar to Equation (4.10), we must do a final dot-product with a spherical harmonic vector evaluated at the view direction. A rotation between local and global coordinate frames must also be performed [71]. We refer readers to the original papers [71, 120] for details of these transformations.

It should be noted that the CPCA method is a combination of two commonly used compression schemes: (global) PCA and vector quantization (VQ) [39] or partitioning the dataset into clusters. This combination has been explored before in the learning community, called VQPCA by Kambhatla and Leen [65, 66] and "local PCA" or "piecewise PCA" in machine learning literature on mixtures of linear subspaces. At the extremes, pure VQ is obtained when only a single constant PCA basis function is used in each cluster, while pure PCA is obtained when only a single cluster is used. Note that PCA is a common technique and widely used including in graphics, while VQ is also well known, and has been used before in graphics for light

field compression [83]. Moreover, there has been previous and concurrent work on block-based PCA [91, 100], DCT based surface light field compression [94], eigentextures for light field compression [101], and various local PCA schemes for surface light fields [20, 148].

The 2003 paper of Sloan et al. [120] provides a unified way of combining VQ or clustering with PCA. The clusters need not even be spatially coherent, and similar transport matrices anywhere on the object geometry can be grouped together. Sloan et al. [120] develop a number of algorithms for CPCA compression. Perhaps the simplest idea to grasp conceptually is to iterate between clustering or VQ, and applying PCA compression or local subspaces to each cluster. In their method, they also include a constant term for each cluster, so they actually find an affine low-dimensional subspace. There are also a number of improvements to redistribute the fitting error, and to optimize cluster coherence for hardware rendering. Finally, the technique is extended to include the effects of subsurface scattering as well [63]. Images showing real-time rendering of glossy materials are shown in Figure 5.3.

Fig. 5.3 Images from Sloan et al. [120], showing real-time rendering of a translucent and glossy Buddha model in two lighting environments.

# 6

# Dynamic Lighting, View, Materials, Geometry

As described so far, the PRT framework allowed for impressive effects under dynamic lighting. However, viewpoint changes were limited to low-frequency or diffuse materials, and objects were assumed to be static. In contrast, offline or standard hardware rendering allows all of the lighting, view, materials, and geometry to vary with time. In the years from 2004 onwards, much work on PRT strived to remove existing limitations and enable at least some of viewpoint, material, and geometry/scene changes. In this section, we describe and survey this literature.

## 6.1 All-Frequency Relighting with Changing View

In 2003, Sloan et al. [120] introduced the CPCA technique, that for the first time enabled dynamic illumination and changing view. However, the method was limited to spherical harmonics, and therefore low-frequency lighting and materials. Simultaneously, Ng et al. [96] had introduced the first all-frequency relighting technique, but it

was limited to images with fixed view, or to diffuse geometry with no viewpoint-dependence. A natural question arose as to whether *all-frequency relighting of glossy objects with changing view* was possible.

The challenge is one of dimensionality. To represent the change in appearance of a surface point over all possible lighting and viewing directions, we need to store a 6D quantity (2D for each of the lighting, view, and surface location). Even with only 100 samples in each dimension, this corresponds to $10^{12}$ samples — too many for most of today's computers in terms of memory and computation speed. Indeed, even the precomputation time could easily take a year or more! Previous methods responded to this in one of two ways. The low-frequency approach of Sloan et al. [120] assumed a low-frequency approximation, with only 25 spherical harmonics for each of light and view-dependence (a total of $625 = 25^2$ basis functions). This corresponds effectively to using only five samples along each light and view dimension ($5^2 = 25$, $5^4 = 625$), which makes the size practical. Alternatively, all-frequency relighting approaches fixed the view, effectively needing to store only a simpler 4D quantity (a 2D image and 2D of lighting variation over the incident sphere of illumination directions).

The challenge is how to enable both dynamic illumination and view, while dealing only with lower-dimensional precomputed quantities. In 2004, three papers [85, 97, 143] addressed this problem. They first perform a factorization of the lighting computation that reduces the core component to (a small number of) 4D problem. The approach of Ng et al. [97] is to factor the light transport into visibility and BRDF terms that are then combined at run-time using the new mathematical idea of wavelet triple product integrals. The approach of Wang et al. [143] and Liu et al. [85] is to factor the BRDF into a sum of terms that depend only on incident and outgoing directions. Each of these terms can be written with a view-independent light transport matrix, analogous to that in [96]. The result is weighted with a view-dependent factor that is independent of the illumination, and summed over all BRDF terms.

### 6.1.1 Triple-Product Wavelet Integrals

We first discuss Ng et al. [97]. They start with Equation (4.1) (limited to direct lighting only) as their basic expression. Because of its widespread use, we reproduce it here

$$B(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega_{4\pi}} L(\boldsymbol{\omega}_i) V(\mathbf{x}, \boldsymbol{\omega}_i) \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}), 0)\, d\boldsymbol{\omega}_i. \quad (6.1)$$

The integrand is a product of the three terms required for direct lighting: the illumination $L$, the visibility $V$, and BRDF $\rho$. It is common in PRT systems to group the visibility and BRDF terms into a single light transport function or matrix — that for glossy objects will be view-dependent, and depend on spatial location and incident direction.

However, Ng et al. [97] *factor* the light transport function, separately considering visibility and BRDF factors. This reduces a 6D function into a product of two 4D forms. The visibility depends on spatial locations $\mathbf{x}$ and incident directions $\boldsymbol{\omega}_i$ (2D in each), and the BRDF on incident and outgoing directions (again 2D in each). Moreover, note that the geometric and material effects have been factored out. The visibility term depends only on scene geometry, independent of reflectance properties, while the BRDF representation depends only on reflectance, not on scene geometry. Thus, it is possible for example, to switch at run-time between multiple (precomputed) BRDFs.

The idea of factoring the light transport is simple and intuitive. The rendering computation is also simple in principle. As shown in Figure 6.1, for each spatial location and corresponding outgoing direction, we have three functions (stored as cubemaps) defined over incident angle — the lighting, visibility, and BRDF. We simply consider the product of these three functions and then add up or integrate over all incident directions.

However, doing the integration with factored forms in real-time is a significant challenge. The basic problem is that instead of the simple summation or dot-product in Equation (4.6), we have what we call a *triple-product integral*. Expanding the angular or $\boldsymbol{\omega}_i$ dependence of lighting, visibility and BRDF in Equation (6.1) in terms of basis

Fig. 6.1 The wavelet triple product method of Ng et al. [97] for all-frequency relighting with changing view. On the left is an image of a scene with approximately 300,000 vertices that can be relit interactively with dynamic view, taking about 3–5 sec per frame. For each vertex on object geometry, we must multiply and integrate three functions over the incident directions (shown as cubemaps on the right) — the lighting, visibility, and BRDF. By factoring the light transport into visibility and BRDF, the method keeps space and computation manageable, while allowing true all-frequency effects. Fast computation is enabled by a new framework of wavelet triple product integrals.

functions $\Psi(\boldsymbol{\omega}_i)$,

$$B(\mathbf{x},\boldsymbol{\omega}_o) = \int_{\Omega_{4\pi}} \left(\sum_j L_j \Psi_j(\boldsymbol{\omega}_i)\right)\left(\sum_k V_k(\mathbf{x})\Psi_k(\boldsymbol{\omega}_i)\right) \times \left(\sum_l \rho_l(\mathbf{n}(\mathbf{x}),\boldsymbol{\omega}_o)\Psi_l(\boldsymbol{\omega}_i)\right) d\boldsymbol{\omega}_i. \quad (6.2)$$

Note that $\Psi(\boldsymbol{\omega}_i)$ could be any orthonormal basis, such as spherical harmonics, wavelets or Fourier series. The coefficients of lighting, visibility, and BRDF, respectively, are $L_j$, $V_k$, and $\rho_l$, which are independent of the angular direction $\boldsymbol{\omega}_i$ and can be taken out of the integral. The lighting is assumed a distant environment map, so coefficients $L_j$ are constant (the technique could also be extended to near-field lighting by

sampling the illumination $L_j(\mathbf{x})$ at a sparse set of spatial locations). The visibility coefficients $V_k(\mathbf{x})$ are a function of spatial location, while the precomputed BRDF coefficients $\rho_l(\mathbf{n}, \boldsymbol{\omega}_o)$ are noted explicitly as a function of both normal and outgoing direction (in practice, a reparameterization by reflected direction is usually performed). Also note that the BRDF coefficients include the cosine term.

We now proceed to simplify the above expression. Since we are focusing on a single spatial location $\mathbf{x}$ (with associated normal $\mathbf{n}(\mathbf{x})$ and outgoing direction $\boldsymbol{\omega}_o(\mathbf{x})$), we drop the dependence on $\mathbf{x}$, $\mathbf{n}$ and $\boldsymbol{\omega}_o$ for clarity. We also move the summations and coefficients that do not depend on $\boldsymbol{\omega}_i$ outside of the integral,

$$\begin{aligned} B &= \sum_j \sum_k \sum_l L_j V_k \rho_l \int_{\Omega_{4\pi}} \Psi_j(\boldsymbol{\omega}_i) \Psi_k(\boldsymbol{\omega}_i) \Psi_l(\boldsymbol{\omega}_i)\, d\boldsymbol{\omega}_i \\ &= \sum_j \sum_k \sum_l C_{jkl} L_j V_k \rho_l. \end{aligned} \tag{6.3}$$

In the expression above, we refer to $C_{jkl}$ as the *tripling coefficients*. For a fixed set of basis functions, they can be usually be obtained analytically by evaluating the integral $\int \Psi_j \Psi_k \Psi_l$, and are a fundamental property of the basis representation used.

Note that the properties of the tripling coefficients can be complicated, and for a general basis, there is no reason for sparsity, i.e., we need to consider all $(j,k,l)$ triplets so the complexity is $O(N^3)$, where $N$ is the total number of basis functions. This stands in stark constrast to the standard *coupling coefficients* and double-product integrals previously used for light transport. In that case, the integral of two orthonormal basis functions is non-zero only when both basis functions are the same, and the complexity is always $O(N)$, reducing to a simple summation or dot-product of lighting and transport coefficients as in Equation (4.6).

Clearly, tripling coefficients and triple-product integrals are more complex — an important contribution of Ng et al. [97] is studying them in detail in a number of different bases. They show that for Haar wavelets, the sparsity (or the number of non-zero coefficients $C_{jkl}$) is $O(N \log N)$, close to the best case linear complexity of the simple pixel basis. Moreover, the Haar wavelet basis is exponentially

better than pixels for compressing complex lighting and visibility signals. For this reason, it is the representation of choice for triple-product integrals.

Besides the basic $O(N \log N)$ sparsity of tripling coefficients in the Haar basis, Ng et al. [97] derive a number of additional interesting results. First, using dynamic programming and the regularity of the tripling coefficients, it can be shown that one can actually derive a linear time $O(N)$ algorithm to perform the summation in Equation (6.3). Moreover, using lazy evaluation, the time can be reduced to $O(n)$, where $n \ll N$ is the number of coefficients actually used, that is usually $0.1\%-1\%$ with wavelet approximation.

The triple-product wavelet integral method can give some stunning results for all-frequency relighting, as shown in Figure 6.1. Note that the technique is not real-time but takes 3–5 sec for relighting and positioning the viewpoint. Fairly complex geometric models of up to 300,000 vertices are demonstrated, and are in fact required to capture the subtleties of the intricate shadowing and glossy reflection effects. The method is most useful in an application, like lighting and view design, to set the illumination and best viewpoint for a scene, where it can speed up the design cycle by an order of magnitude.

Finally, it is worthwhile noting that the triple-product formulation is very relevant to another fundamental problem, that of multiplying two signals, each represented in an orthonormal basis such as Haar wavelets [97]. This has many potential applications in multimedia and signal-processing. In PRT itself, products of functions are important for accumulating visibility from multiple blockers in the shadow field method for rendering dynamic scenes [150], described later in this section. One immediate application in offline rendering is to wavelet importance sampling for the product of illumination and BRDF [23].

### 6.1.2 BRDF In–Out Factorization

The triple-product method factors light transport by separating the visibility and the BRDF. In concurrent work in 2004, Wang et al. [142] and Liu et al. [85] introduced an alternative factorization of the BRDF

into terms depending only on incident and only on outgoing angles,

$$\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \approx \sum_{k=1}^{K} \sigma_k h_k(\boldsymbol{\omega}_o) g_k(\boldsymbol{\omega}_i), \tag{6.4}$$

where $h_k$ and $g_k$ are 2D maps. In practice, this form is obtained from a SVD of the BRDF, and $\sigma_k$ are the corresponding singular values or eigenvalues obtained (denoting the importance of various terms in the factorization). The effect of increasing the number of terms $K$ is shown in Figure 6.2.

This factorization allows us to define and precompute a view-independent transport function for each factor $k$ in Equation (6.1),

$$T_k(\mathbf{x}, \boldsymbol{\omega}_i) = V(\mathbf{x}, \boldsymbol{\omega}_i) g_k(\boldsymbol{\omega}_i) \max(\boldsymbol{\omega}_i \cdot \mathbf{n}(\mathbf{x}), 0). \tag{6.5}$$

Rendering then reduces to

$$B(\mathbf{x}, \boldsymbol{\omega}_o) = \sum_{k=1}^{K} \sigma_k h_k(\boldsymbol{\omega}_o) \int_{\Omega_{4\pi}} L(\boldsymbol{\omega}_i) T_k(\mathbf{x}, \boldsymbol{\omega}_i) \, d\boldsymbol{\omega}_i. \tag{6.6}$$

Fig. 6.2 This figure, from Wang et al. [145], shows a progression from a gray diffuse BRDF on left, to including more BRDF in–out factors ($K = 1$ in the middle, and $K = 4$ on right). As can be seen in the figure, moderate specularities can be captured even in the middle image, and real-time rendering is possible with dynamic lighting and view. The BRDF in these figures is the anisotropic Ashikhmin-Shirley BRDF [8].

The integral can now be evaluated by standard PRT methods, since it is view-independent and reduces to essentially the same form as Equation (4.6). In fact, for each term $k$ separately, it can be written simply as $\mathbf{T}_k(\mathbf{x}) \cdot \mathbf{L}$, a dot-product of a column-vector of lighting and transport wavelet coefficients for each vertex $\mathbf{x}$. This is similar to diffuse all-frequency radiance transfer, since the result is view-independent. All that remains is the modulation of the integral by the view factor $h_k(\boldsymbol{\omega}_o)$, and to weight by $\sigma_k$ and sum over all terms $k$.

The conceptual idea is straightforward. The remaining items involve compression. The transport matrix $\mathbf{T}_k$ for each term $k$ can be compressed spatially using CPCA, in much the same way as used in Sloan et al. [120]. A more recent innovation is to leverage work in tensor analysis, making use of Clutered Tensor Approximation (CTA) [133]. In addition, as with most all-frequency methods, we make use of wavelet representations of the lighting and similarly wavelet transform the rows of the transport matrices. The details of the particular implementation can be found in [85, 142]. Both papers demonstrate real-time all-frequency relighting of scenes containing glossy objects, allowing simultaneous manipulation of illumination and viewpoint. Moreover, the technique has become popular enough that it has been used as a basic building block in many subsequent PRT algorithms [59, 87, 102, 129, 133, 144, 145, 146].

### 6.1.3 Discussion

Both wavelet triple products and BRDF in–out factorization have significant benefits, but also limitations and tradeoffs. The main advantage of the triple product method is that it supports true all-frequency effects — each of the lighting, visibility, and reflectance can have very high frequencies, including sharp highlights and shadows. However, signal compression like clustered PCA is not possible across vertices of the geometry. This is because visibility and BRDF must be combined at run-time, and they are represented in different spaces — visibility spatially over the vertices of the mesh, and the BRDF in the angular domain over orientations. Any scheme to exploit spatial coherence must consider these two fundamentally different domains simultaneously, and to date no progress has been made. For this reason, large datasets

are needed, on the order of those in the original all-frequency relighting paper [96]. Moreover, rendering remains expensive taking a few seconds a frame, with true real-time performance still out of reach.

The main benefit of the BRDF in–out factorization approach is its speed and simplicity, with many real-time demonstrations in the original and subsequent papers. However, the method scales linearly in both space and time with the number of BRDF terms $K$, and most work has used a very small $K$, usually less than 10. Recently, Mahajan et al. [88] have formally analyzed the accuracy of the in–out BRDF factorization, and shown that for a Phong BRDF, the number of terms needed scales approximately as $2.5s$, where $s$ is the Phong exponent. Thus, even a moderate Phong exponent with $s = 30$ requires 75 terms, that is too much for current systems. In effect, only broad specular lobes or low-frequency BRDFs can be represented. Note that there are other BRDF factorizations like half and difference angles [115] that allow for many fewer BRDF terms. But those representations do not allow the transport functions $T_k$ to be view-independent as required. Given these limitations, a number of alternative approximations and factorizations have been proposed [43, 44] for practical purposes, that may provide the best real-world solution for some cases.

Finally, the nature of the triple product method, with a factorization of visibility and BRDF, precludes view-dependent global illumination effects, and the method has been restricted to direct lighting only. This was also a restriction in the original papers that used BRDF in–out factorization, but subsequent work by Wang et al. [145] has developed an elegant iterative method to build up transport matrices for further interreflection bounces, so global illumination can now be included.

In summary, wavelet triple products and BRDF in–out factorization have made significant steps towards the goal of all-frequency relighting of glossy objects with changing view. However, in the author's view, this fundamental problem in real-time rendering is still in need of a better solution. The space and time efficiency of triple products is severely limited by the inability to incorporate spatial compression techniques like CPCA. The BRDF in–out factorization is limited in that high-frequency materials can require a hundred or more BRDF terms, making the method impractical for real-time use in these cases.

## 6.2 Dynamic Materials and BRDF Editing

The work on PRT described thus far has focused on allowing dynamic manipulation of lighting and later viewpoint. Besides scene geometry, object reflectance properties are held fixed. Note that some variation in BRDFs is possible — in the original method of Sloan et al. [121], low-frequency Phong materials could be specified at run-time, while Ng et al. [97] allow the user to swap in a set of discrete precomputed BRDFs, each of which can occupy significant storage. However, none of these methods were intended or allow for continuous *BRDF editing*; wherein complex realistic materials can be manipulated in natural lighting for the user to determine the best set of reflectance parameters.

Indeed, reflectance editing and design is as important for specifying scene attributes as lighting design. However, previous systems developed for BRDF editing usually allowed only for simple lighting. A typical system allowed the user to adjust analytic parameters while visualizing the results in a simplified setting such as an unshadowed point light source. However, this does not allow one to gauge the subtleties of appearance with objects in their final placement in a real scene, with realistic illumination. A natural question is then whether PRT methods for complex lighting can be adapted to *real-time BRDF editing in complex lighting*. This problem was addressed by Ben-Artzi et al. [14], whose solution we briefly describe.

Just as standard all-frequency PRT fixes all the attributes except illumination, the algorithm in Ben-Artzi et al. [14] fixes the scene, lighting and viewpoint, but allows for general all-frequency edits to reflectance properties (BRDFs), as shown in Figure 6.3. The main idea is to start from Equation (6.1), but write the BRDF as an expansion in terms of basis functions, instead of the lighting. Conceptually, this leads to the same result as in relighting, with the final image a wavelet dot-product of generalized transport and BRDF coefficients. The generalized "transport" is now obtained by the product of lighting and visibility (rather than BRDF and visibility).

The challenge is that the BRDF is a 4D quantity, not easily amenable to a compact basis expansion in spherical harmonics or

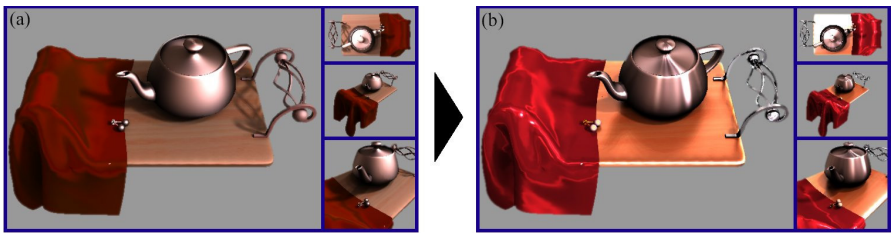


Fig. 6.3 Results of real-time BRDF editing, from Ben-Artzi et al. [14]. The scene is illuminated by an environment map; the main view is augmented by small number of additional views on the right to provide more view-dependent feedback. The cloth and handles use measured BRDFs, while the other objects use various analytic models, showcasing the generality of the system. The analytic parameters of the teapot BRDF model are adjusted to make it more anisotropic, while the tray is made more specular and freehand edits are made to the measured reflectance of the cloth to create novel artistically-designed BRDFs.

wavelets. Ben-Artzi et al. [14] address this by noticing that the variable part of most existing analytic or measured BRDFs can be encapsulated in a single (or at most two) 1D functions,

$$\rho(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) \equiv \rho_q(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) f(\gamma(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)) = \rho_q(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) \sum_{j=1}^{J} c_j b_j(\gamma), \qquad (6.7)$$

where $\rho_q(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)$ is a general but fixed (uneditable) part of the BRDF, such as the shadowing terms in the Torrance-Sparrow BRDF. $f(\gamma(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o))$ encapsulates the editable part of the BRDF as a 1D function or curve. For analytic BRDFs like the Phong model, it is controlled by analytic parameters like the Phong exponent. Here, $\gamma(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)$ is a suitable parameterization, such as the reflected direction (Phong) or half-angle direction (Blinn-Phong or Torrance-Sparrow). The basis functions $b_j(\gamma)$ are now in 1D, and can be wavelets or simple box functions. For analytic BRDFs, they represent a linearization of the BRDF as needed for PRT (most BRDF models are highly non-linear in their intrinsic parameters). For measured reflectance, the function $f(\gamma)$ can be directly edited using a variety of curve editing tools.

The "transport" function for fixed lighting and view is then given for each spatial location $\mathbf{x}$ or pixel in the final image by

$$T_j(\mathbf{x}) = \int_{\Omega_{4\pi}} L(\boldsymbol{\omega}_i) V(\mathbf{x},\boldsymbol{\omega}_i) \rho_q(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o(\mathbf{x})) b_j(\gamma(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o(\mathbf{x}))) \, d\boldsymbol{\omega}_i, \qquad (6.8)$$

where we note that for fixed lighting and view, the outgoing direction $\boldsymbol{\omega}_o(\mathbf{x})$ is a deterministic function of $\mathbf{x}$. Finally, just as in image relighting, rendering for BRDF editing can be written as

$$B(\mathbf{x}, \boldsymbol{\omega}_o(\mathbf{x})) = \sum_j c_j T_j(\mathbf{x}), \tag{6.9}$$

where $c_j$ are the coefficients of the curve being edited and $T_j$ are the transport coefficients. There are a few further subtleties peculiar to BRDF editing. Since we directly visualize the BRDF on objects as opposed to the lighting (which only contributes indirectly to the scene), smoother wavelets than Haar are needed, and Daubechies wavelets [27] can be used. Furthermore, an incremental rendering technique is described to take advantage of frame-to-frame coherence and avoid updating all the coefficients at every time step, providing a speed up of upto an order of magnitude — a refinement of this idea is also relevant to standard all-frequency relighting with fixed BRDFs [102]. Finally, some BRDFs can vary along two parameterizations like half-angle and difference-angle [115]. In this case, the vector of coefficients $T_j(\mathbf{x})$ is replaced by a matrix, much as view-dependent glossy materials are handled by matrices in low-frequency PRT [121].

In subsequent work, Ben-Artzi et al. [13] described an extension to include multiple bounce global illumination and not just direct lighting. This extension is much more difficult for BRDF editing than for relighting. One primary issue is that the final image is not linear in the BRDFs, as it is for relighting. In multiple bounce reflections, the BRDFs of the corresponding surfaces are multiplied together, leading to polynomials with degree corresponding to the number of bounces. This can be seen directly from Equation (2.4) for the rendering equation. The final expression $B = (\mathbf{I} - \mathbf{KG})^{-1}E$ is linear in the incident lighting or emission $E$. However, when the BRDF is introduced explicitly in defining the $\mathbf{K}$ operator, we obtain for BRDF editing,

$$B = E = \mathbf{K}(\rho)\mathbf{G}E + \mathbf{K}(\rho)\mathbf{G}\mathbf{K}(\rho)\mathbf{G}E + \cdots \tag{6.10}$$

Since the reflection operator $\mathbf{K}(\rho)$ is bilinear in the scene BRDFs and the lighting, each bounce corresponds to a polynomial in the BRDF coefficients $b_j$. Ben-Artzi et al. [13] derive the relevant forms

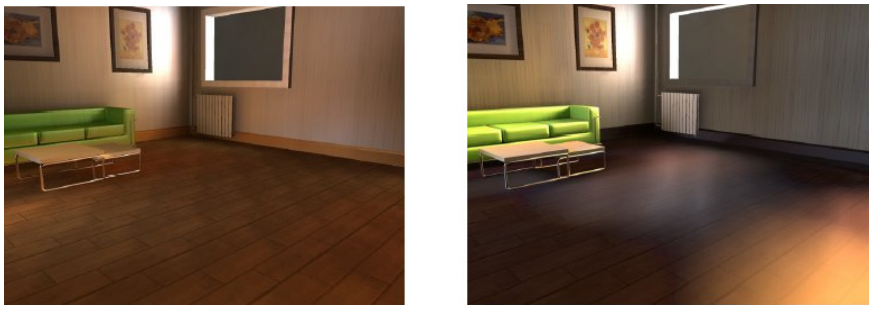

Fig. 6.4 interactive relighting with dynamic BRDFs from Sun et al. [129]. The left and right images show different reflectance properties for the sofa and floor, where the user has adjusted the BRDFs. The images are rendered in real-time with two-bounce interreflections.

directly from the rendering equation, and describe a novel non-linear PRT framework, using a precomputed polynomial representation and a multi-bounce *tensor* of transport coefficients (rather than a simple vector or matrix). Approximations, such as reducing the number of coefficients for BRDFs further from the viewing path, are used to keep the computation tractable. The overall system demonstrates some of the first results of interactive BRDF editing with global illumination.

Finally, the question is whether we can relax the assumption of fixed view and lighting to enable interactive manipulation of all three of: illumination, viewpoint, and material properties. Such a capability could also have immense benefit for including recently developed models of dynamic time-varying BRDFs [47, 127, 138] in interactive applications. Clearly, this is too expensive if we want to preserve full generality and fidelity in lighting and materials. However, if we are willing to make some approximations, such as limiting ourselves to two-bounce reflections and using the in–out BRDF factorization, Sun et al. [129] have demonstrated an effective technique for interactive relighting with dynamic BRDFs. Example images are shown in Figure 6.4.

## 6.3 Dynamic Scenes and Deforming Geometry

We have seen extensions to the basic PRT framework that allow variation in lighting, view, and materials. We now consider the critical limitation of PRT to static scenes. A large body of work in the period

2004- has tried to relax this restriction. Note that the solutions are not entirely general — the reflectance is fixed, while dynamic geometry is usually limited to rigid objects moving with respect to each other, or to local deformations of a surface. The ability to combine local and global geometry changes, such as a character deforming their arm and simultaneously casting a shadow onto the torso, is still limited.

The initial papers on dynamic scenes in PRT were by Mei et al. [92] and Kautz et al. [69]. These approaches can be seen effectively as using precomputed data structures to speed up standard brute-force ray tracing and rasterization, respectively. Mei et al. [92] precompute the information for ray-model intersection using a spherical radiance transport map, on the bounding sphere surrounding the object. After that, essentially standard ray tracing can be used, but with the cost of ray-surface intersections reduced to ray-bounding sphere intersection. Since the lighting is sampled into point or directional sources [1], near-field area lights can also be handled relatively easily. The method in Kautz et al. [69] is based on the low-frequency spherical harmonic PRT approach in Sloan et al. [121]. It observes that the bottleneck to dynamic geometry is precomputation, and develops fast approximate techniques to rasterize the entire occlusion information over the sphere for (self-) shadowing at each vertex at run-time.

These direct approaches have had some success. Moreover, recent techniques have shown effective ways to interactively compute simpler attributes like ambient occlusion [75, 76]. However, the most popular PRT methods for dynamic scenes are two papers from 2005, on local deformations by Sloan et al. [123], and on precomputed shadow fields for motions of rigid objects by Zhou et al. [150]. We now proceed to describe these methods in detail.

### 6.3.1 Local Deformable PRT

Consider a motion such as a bat flapping its wings, as in Figure 6.5. The light transport for any vertex will change due to the deformation of the geometry. The key observation is that locally, the geometry is simply rotating, and if we apply the corresponding rotation to the transport function, we can still integrate against the lighting as in standard PRT.

Fig. 6.5 Local deformable PRT [123] for a bat flapping its wings. The single lobe zonal harmonic model at each point is rotated in real-time, to create these images rendered at 240 Hz, including diffuse PRT and subsurface scattering.

However, this method does not capture global effects such as the change in shadowing on the torso of the bat as the wings flap.

For low-frequency PRT, rotation in spherical harmonics is actually a relatively straightforward operation, which is one reason they are so popular as a basis representation. In particular, a spherical harmonic $Y_{lm}$ transforms to other spherical harmonics $Y_{lm'}$ with the same major index or frequency $l$. The rotation can be expressed with rotation matrices $D^l_{mm'}$ that tells us how much of the energy in $Y_{lm}$ is transported to $Y_{lm'}$ for a particular rotation. However, note that this calculation now needs to be performed at every vertex, since each vertex can have a different rotation as the object deforms. The cost of standard spherical harmonic rotation is $O(N^{3/2})$, where $N$ is the total number of basis functions (usually 25) and is therefore still expensive, especially since it needs to be performed at every vertex. The question is whether we can come up with a faster $O(N)$ algorithm.

The key contribution in Sloan et al. [123] is to derive a new representation of the transport function using zonal harmonics, that enables fast rotations. The zonal harmonics are simply the spherical harmonics $Y_{l0}$. We also add an additional degree of freedom to define the orientation or central direction (rather than centering about the north pole). It can be shown (as seen in Figure 6.6) that a small number of suitably chosen and oriented zonal harmonics suffice to fit complex transfer functions — this idea has subsequently been extended to spherical radial basis function approximations [133]. Of course, the fitting is now not simply a projection, and a suitable optimization process must be used.

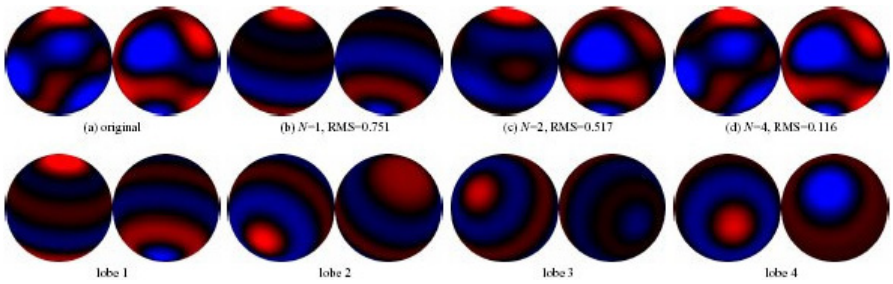


Fig. 6.6 Sums of zonal harmonics can accurately match difficult signals such as the randomly chosen order 6 spherical harmonic approximation in top left (red is positive, blue is negative). The top row shows the increasing quality of approximation with more zonal harmonic lobes, while the bottom row shows the individual lobes. In this case, 4 lobes suffice; often a single zonal harmonic lobe is a good approximation to the light transport function.

The key benefit of zonal harmonics for local deformable PRT, is that rotations are easy to perform. Rotating a zonal harmonic corresponds simply to rotating its central direction. The final projection into global spherical harmonic coefficients then just becomes

$$g_{lm} = Y_{lm}(\mathbf{C}) f_l, \tag{6.11}$$

where the $g_{lm}$ are the desired spherical harmonic coefficients in global coordinates for zonal harmonic $Y_{l0}$ centered at $\mathbf{C}$, and $f_l$ is the coefficient for that zonal harmonic. Of course, this needs to be accumulated over all lobes and all $l$. This is a simple approach, and only 4–6 lobes are usually needed. This work has also spawned interest in corresponding efficient rotations of wavelets by Wang et al. [142], which is a significantly harder problem because wavelets are not designed for ease of rotation and affine transformations.

### 6.3.2 Precomputed Shadow Fields

An alternative approach, suitable when multiple rigid objects are moving with respect to each other is the precomputed shadow field method, first introduced by Zhou et al. [150]. Like Mei et al. [92], the shadowing information is precomputed, i.e., at each point in a volume surrounding an object, we store spherical harmonic or wavelet coefficients for the visibility map, as shown in Figure 6.7 left.

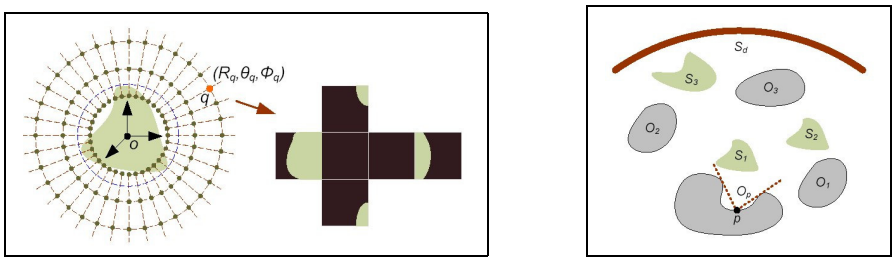


Fig. 6.7 The shadow field method from Zhou et al. [150]. *Left*: Shadow field. Left shows the radial and angular spatial sampling; the field is precomputed at the points shown in concentric shells around the object and interpolated. Right shows the angular cubemap for visibility for one spatial location. *Right*: Combining shadow and source radiance fields for shading. Point $p$ has self-visibility or occlusion $O_p$ and is also occluded by the other three blockers in the scene, $O_1$, $O_2$, and $O_3$. Their visibility contributions will be accumulated. Similarly, contributions from the local light sources $S_1$, $S_2$, and $S_3$ will be combined with that from the distant environment map $S_d$. All of these objects and sources can move rigidly and change overall scene configuration, with the images still being rendered in real-time.

Therefore, for any point in space, we know what the visibility field for the object is, and translating the object simply applies a corresponding translation to where in the map we look up the visibility. Rotations can also be addressed in spherical harmonics, but are harder in a wavelet basis. Note that light sources with complex shapes and angular distributions can also represented in this way, in what the authors refer to as a *source radiance field* — in this case, each point in space holds the angular distribution of outgoing light. Unlike light fields [42, 83] that are 4D quantities parameterized on the space of rays, shadow fields are full 5D quantities that are functions of spatial location in a volume, and angular direction. Storage is therefore a significant issue, but there is enough coherence that a sparse sampling is possible [150]. A theoretical analysis of the exact sampling rate and frequencies needed remains a subject of future work. It is also an open question whether the full 5D representation is needed — while it is very convenient for rendering, all of the information therein can really be obtained as a 4D shadow field, similar to a light field (however, the 4D representation is cumbersome for real-time lookup and rendering).

We can simply accumulate the shadow fields for all the occluders to obtain the shading at any vertex, as shown in Figure 6.7 right. In particular, the net visibility is given by the *product* of each individual

visibility map. We have already seen how products of functions can be computed efficiently directly in the spherical harmonic or wavelet domain using tripling coefficients and the framework of triple-product integrals [97]. We simply need to multiply two visibility functions, then use the product to multiply other terms and so on. Finally, a standard dot-product (double product integral) of lighting and net visibility (transport) coefficients can be performed. For glossy objects, the final step can be a triple-product integral of lighting, net visibility and BRDF. Multiple area sources can also be included using source radiance fields. An example image from the method is shown in Figure 6.8.

The shadow field method is a simple elegant idea that enables soft shadows in dynamic scenes. Subsequent years have seen a number of extensions and improvements. The core computational problem is a product of multiple visibility maps from shadow fields. While all-frequency effects can be obtained by multiplying them two at a time using wavelet triple-products, this is not the most efficient approach. Sun and Mukherjee [128] have shown how multifunction product integrals can be efficiently evaluated in wavelets to enable a host of new techniques for all-frequency relighting of dynamic scenes. For spherical harmonics, the triple-product can be slow. Ren et al. [113] develop

Fig. 6.8 Rendering of soft shadows in a dynamic scene with shadow fields [150]. The objects are moving, as is the lighting from both local light sources and a dynamic environment map.

an intriguing idea of working in the log domain so that multiplication simply becomes an addition of spherical harmonic log-visibility maps. New mathematical approaches to efficiently compute the final spherical harmonic exponential needed are then described. The method of Ren et al. [113] is also interesting in that blockers are approximated by sphere sets, with limited level of detail control. Since the sphere sets can be computed at run-time, no precomputation is needed and fairly general scenes can be incorporated.

Finally, many of the newer PRT methods are optimized for environment maps and other direct lighting configurations. The shadow field work is no exception and the original paper [150] did not consider global illumination. However, recent papers [59, 146] also include interreflections in dynamic scenes, by precomputing the impact of basis functions for irradiance over the blocker object, and then computing their effects on the receiving surfaces.

## 6.4 Discussion and Limitations

The generality of PRT methods has come a long way from the original paper by Sloan et al. [121]. All-frequency effects can be included, not just spherical harmonic lighting. Moreover, not only lighting, but also viewpoint can be adjusted interactively. Of course, the holy grail of real-time rendering is to go beyond lighting and view variation, and also allow dynamic scenes and changing geometry, as well as BRDF editing and time-varying reflectance. In the extreme case, the scene is completely different from frame-to-frame and no coherence can be exploited for precomputation. This is basically the approach taken in hardware rendering — but at the cost of many realistic effects — or in software rendering, at the cost of real-time performance.

The idea of precomputed radiance transfer is to find a middle ground, assuming there are some quantities that do not change over the course of the real-time rendering (such as static geometry). This enables one to precompute relevant information, followed by efficient processing to enable real-time rendering in dynamic lighting. In the most recent PRT methods, lighting and view can be varied in real-time, but either the reflectance or geometry must be fixed. (Since many

degrees of freedom must be precomputed, we do not see an easy path to varying all four quantities in PRT algorithms.) Moreover, dynamic scenes are limited to either local deformable PRT or shadow fields for rigid objects. One cannot combine local and global deformations easily. Some recent work [61, 134] has tried to combine animation and simulation information with shading, but as the number of animation modes (there can be 40+ joint angles for a human character) increases, the amount of precomputed data grows exponentially. Thus, dynamic scenes remain a big challenge for PRT.

In summary, the PRT technique and generalizations have brought a host of new effects into the real-time domain for the first time, and enabled a high level of generality. However, the final challenge for real-time rendering remains — fully dynamic lighting, view, materials, and geometry, with all-frequency shadows and global illumination.

# 7

# Extensions

So far, we have covered the core PRT methods. In this section, we briefly discuss a number of ways in which PRT ideas and algorithms have been used and extended to enable a variety of novel effects. Specifically, we briefly describe PRT methods for lighting design with global illumination, the inclusion of local lights in PRT, volumetric effects and subsurface scattering, new theoretical advances, and mathematical methods based on PRT but that do not require precomputation. Examples of some of the images produces by these extensions to PRT are shown in Figure 7.1.

Fig. 7.1 Extensions to PRT to enable a wide range of effects. *Left*: PRT extended to enable direct-to-indirect transfer for cinematic relighting [52]. *Middle*: Affine double and triple-product wavelet integral computations to allow propagation of light from arbitrarily shaped and textured area sources, enabling near-field lighting [125]. *Right*: Translucent materials with single and multiple scattering rendered with environment map lighting [144].

One major challenge for film production and animation is lighting design, i.e., setting up the illumination for a synthetic computer-generated scene. This is a manual iterative process, and we would like the iterations and feedback to be interactive. But each iteration has historically taken minutes to hours, as the scene needs to be re-rendered in a slow offline process for each modification to the lights. Note that the view, geometry, and BRDFs are typically fixed in this application, with only the lighting being modified. This makes lighting design very amenable to precomputation-based methods for acceleration. Unfortunately though, the base PRT method is specialized for environment map illumination, and usually does not have the generality to handle the range of local light sources typically used by a studio for lighting design. Moreover, much of the work on PRT has focused on direct lighting, while movie production is increasingly incorporating interreflections and global illumination.

Kristensen et al. [78] developed the first full lighting design system using PRT. They addressed local lighting by using unstructured light clouds, which enables lighting design with global illumination and near-field sources. Subsequent work by Lehtinen et al. [81] provides a more efficient precomputation using meshless light transport. Hasan et al. [52] made the observation that the direct lighting can often be computed using modern graphics hardware techniques [38, 104, 107], and with fairly general light sources and material shaders. The challenge then is for indirect illumination, and they develop a PRT technique for "direct to indirect transfer." In other words, given the direct lighting, they compute the result of interreflections, or the full global illumination solution. This is a variation of PRT where instead of a distant environment map, we have the incident direct radiance in a scene. Note that the final global illumination solution is still linear in this incident radiance, and we simply need to precompute the light transport matrix for the scene. In simultaneous work, Kontkanen et al. [77] explored a finite element wavelet-based technique for real-time indirect lighting; fast indirect reflections were also studied by Hill and Faloutsos [57].

Another significant challenge for PRT systems has been local lighting, with much of the focus on distant environment maps. The original

paper by Sloan et al. [121] did include the idea of resampling the lighting at a sparse set of locations on the object. However, using simply 8–10 locations do not suffice to capture sharp spatial variations and local lighting. This remains a major obstacle to widespread use of PRT in interactive applications, but some progress has been made in recent years. Annen et al. [5] introduced the idea of spherical harmonic gradients to somewhat reduce the spatial sampling rate for mid-range lighting. A related idea is the notion of spherical harmonics scaling by Wang et al. [139]. However, near-field (rather than mid-range) or all-frequency lights cannot be easily handled. A more brute-force approach is the notion of source radiance fields by Zhou et al. [150] (or the related spherical radiance transport maps of Mei et al. [92]). This method can encapsulate any light source but at the cost of significant storage of a 5D volume. Also, the content of the lighting is fixed at precomputation and cannot be edited (such as changing its texture) at run-time. Other work by Ren et al. [113] allows for (only) spherical area sources. Recently, Sun and Ramamoorthi [125] have developed an extension of wavelet double and triple-product integrals that allows one to transport lighting directly from a local textured planar area light source using affine transforms. Overall, while significant progress has been made, local lighting remains a challenge and future work direction for PRT.

Most computer graphics scenes, and essentially all of the PRT algorithms discussed so far, have assumed clear day conditions and opaque objects. Wang et al. [144] develop a simple PRT method and fast precomputation for rendering objects with subsurface effects, including single and multiple scattering. In later work, they enable real-time editing and relighting of homogeneous translucent materials [141]. Sun et al. [126] show how to easily include atmospheric single scattering effects in a PRT system, so we can see how an object looks like on a misty or foggy day. PRT methods have also been extended for real-time rendering of plant leaves [140] including BRDF, subsurface, and transmission effects.

Besides being a practical algorithm to enable a range of new real-time effects, PRT has led to a new deeper understanding of the theoretical and computational aspects of the reflection and rendering equations.

The wavelet triple-product method [97], with recent extensions to affine forms for wavelets [125] and spherical harmonics [139], also makes significant mathematical contributions to numerical techniques that are likely to be relevant in other fields. Moreover, the theoretical foundations of PRT, and their relation with conventional global illumination rendering, have been elucidated by Lehtinen [79]. The mathematical foundations of locally low-dimensional approximations like CPCA have also been studied [87], as has the error in BRDF factorization [88]. Many of the mathematical ideas introduced by PRT have also been applied to real-time rendering problems that do not directly require precomputation. One example is the spherical harmonic exponential technique for soft shadows [113]. While it is based on PRT methods for dynamic scenes and the shadow-field method [150], precomputation is avoided by dynamically creating a sphere set representation for the geometry. Many newer mathematical PRT ideas like Spherical Radial Basis Function (SRBF) representation [133] and frequency domain convolution have also recently been used by Han et al. [51] for addressing the long-standing problem of normal map filtering.

PRT continues to be an active area of research. In the past few years, entire sessions at major conferences like SIGGRAPH and the Eurographics Symposium on Rendering have been devoted to the topic. Some examples of newer features in PRT include the ability to incorporate local surface detail like bi-directional texture functions [26] — Sloan et al. [122] show how to do so using bi-scale radiance transfer for the coarse geometric scale and the fine local microstructure. Sloan [119] also develops methods to include normal mapped objects in PRT. New basis functions for PRT are also still being investigated, and non-linear cut-based techniques, that impose hierarchies on more unstructured data, have been used successfully for combined lighting and material design [3, 22].

Finally, the success of spherical harmonic-based PRT algorithms in industry for video game production and other interactive applications, has spawned a wide range of tutorials and software. Courses on PRT have been taught at SIGGRAPH and the Game Developer's Conference. An excellent tutorial on spherical harmonic lighting is by Green [45], and an article on spherical harmonic tricks by Sloan [118].

# 8

# Future Research

In this section, we look at some of the current limitations of PRT systems, and areas that are ripe for future research. Specifically, we discuss practical all-frequency effects, the relationship to modern graphics hardware methods, fast precomputation algorithms, dynamic scenes, PRT for design applications, and the idea of precomputation in other areas of graphics.

The core problem that PRT seeks to address is real-time rendering of photorealistic images with global illumination, under dynamically varying lighting and view. However, current practical adoption is usually limited to spherical harmonic low-frequency techniques. The all-frequency methods introduced by Ng et al. [96] enable a richer range of effects but involve large storage costs and are cumbersome to implement in graphics hardware. Moreover, for all-frequency effects, simultaneous manipulation of lighting and view is difficult. As discussed earlier, wavelet triple-products [97] are limited to direct lighting and do not enable taking advantage of spatial coherence, while BRDF factorization approaches [85, 143] limit the use of high-frequency materials. Therefore, true all-frequency effects remain a challenge in PRT. It is also possible that the solution here may involve a combination

of existing techniques, rather than a new PRT method. One possibility is an approach that breaks the environment into a sparse set of directional lights plus low-frequency spherical harmonics. Another possibility is the use of new representations such as non-linear cuts [22] or spherical radial basis functions [133].

One of the benefits of the original PRT method [121] was that it reduced the real-time rendering problem to matrix multiplications and vector dot-products, that could be easily performed in the new generation of programmable graphics hardware. As the flexibility and speed of graphics hardware has grown, a natural question is whether a precomputation-based approach is needed at all. Clearly, we would prefer to avoid precomputation and provide more flexibility for dynamic scenes if possible. Currently, environment maps can often be reduced to a few hundred light sources [1] for offline rendering. If shadow-mapped rendering from these sources can be performed at interactive speeds, all-frequency effects can be achieved without any precomputation. Similar arguments hold for the new generation of fast ray-tracing systems. A recent paper in this direction is by Annen et al. [4], that improves on conventional shadow mapping, while allowing real-time performance with at least 30 light sources representing an environment map. Even caustics can be rendered in real-time today without precomputation, by explicitly tracing photons [130]. Thus, we believe it is entirely possible that a non-PRT approach could achieve many of the effects in the previous sections in real-time. However, we predict that more complex interreflection and global illumination effects may still need precomputation for real-time performance.

While much research in PRT has been focused on real-time rendering, the precomputation is a crucial step and accelerating it will enable greater flexibility and ease of use. For direct lighting, the hardware-based method of rasterizing the visibility at each vertex [96] gives excellent performance even for all-frequency shadowing, so precomputation times can be reduced to a few minutes. Nevertheless, it remains an open question if visibility coherence can be exploited to further speed up precomputation — and in the limit allow the precomputation itself to run in close to real-time for dynamic geometry. This is even more interesting in the context of global illumination. Clearly, there is a lot

of coherence in the light transport matrix, with many PRT methods in fact relying on exploiting spatial and angular domain coherence. If we are going to finally use a heavily compressed representation, can we avoid needing to generate the full light transport, and create the compressed representation directly or from a sparse set of samples? The matrix row–column sampling work [53] and recent efforts in compressive light transport sensing [103] are two techniques for offline rendering and acquisition that may inspire similar approaches for PRT precomputation.

Dynamic scenes and changing geometry remain a difficult challenge for PRT. Limited possibilities to local deformations [123] and rigid moving objects [150] exist. But general animations or motions of characters in a video game remain challenging. One approach would be to precompute a higher-dimensional space including as axes the various degrees of motion. A first step, for ambient occlusion, has been taken in [75]. But this quickly becomes impractical for complex visual effects and human-scale motion. Interesting possibilities may be to keep track of events where shadowing or interreflections change, or to work with low-dimensional representations of motion captured data. In any event, the core PRT techniques are limited to static scenes, and finding the right representations for dynamic characters remains a difficult problem.

The main application of PRT has been to create a new set of visual effects in interactive applications. Recent work has also explored its use as an effective tool in lighting design. But there are also other applications of design where precomputation is very useful. We have recently seen interest in PRT-based methods for reflectance design [13, 14, 22]. More interesting appearance design applications like BTF design [68] may be the next steps. Moreover, these design problems are not limited to rendering. The idea of precomputation-based design can be used in physical simulation and animation as well, with many key insights carrying over. More broadly, James and collaborators [10, 61] have shown the benefits of precomputation for fast physical simulation, and we predict a rich range of applications for precomputation-based methods in computer animation and other fields of graphics beyond rendering.

# 9

# Conclusion

One of the ultimate challenges of computer graphics is real-time photo-realistic rendering, with the ability to change the lighting, view, material properties of objects, or scene geometry interactively. Having this capability would greatly improve the realism of virtual environments and interactive applications, allowing the creative intent of content authors to reach their full potential.

Historically however, the goals of realism and real-time have been in severe conflict with many unavoidable compromises in accuracy for real-time performance. Precomputation-based rendering has provided a new and exciting path to bridge the high quality offline world with the real-time domain. We assume some aspects of the scene can be fixed and precomputed, to allow rapid variations in other aspects like dynamic lighting. This survey has gone back to some of the early origins of precomputation-based rendering in replaying offline solutions for image relighting. Following this, we have described the new innovations in terms of environment maps and spherical harmonic convolution, that inspired the PRT method of Sloan et al. [121]. That approach for the first time brought precomputation-based ideas into the mainstream of real-time rendering. The subsequent sections have discussed a number

of new innovations in this field over the past six years, to all-frequency effects, changing lighting and view, reflectance editing, dynamic scenes, and lighting design. These advances have spawned a number of tutorials and software implementations, finally leading to rapid adoption of spherical harmonic lighting by industry.

At this point, PRT has established itself as one paradigm to produce a wide range of realistic effects for real-time rendering. Moreover, many of the theoretical and conceptual advances of PRT have been worked into other areas of rendering, graphics, and other fields. However, the ultimate quest for realism in real-time continues. Whether that approach will be based primarily on PRT, real-time ray tracing, or further advances in programmable graphics hardware remains unclear at this time. We feel it is very likely to be some combination of the three. This remains a time of great excitement in the field, with the quest for photorealistic real-time rendering beginning to fall within reach. The next few years and decade will see great strides in that direction, and precomputation-based methods will remain an important stepping stone towards that ultimate goal.

# Acknowledgments

I would like to thank all of my collaborators over the years. I started to work on spherical harmonics and real-time rendering during my PhD at Stanford University with Pat Hanrahan, to whom I owe a significant debt. My first PRT papers were in collaboration with Ren Ng, who coined the term "all-frequency" and provided the main motivation towards reaching that goal. Subsequent collaborators on the PRT research projects below have included a number of graduate students: Aner Ben-Artzi and Kevin Egan (BRDF editing), Nolan Goodnight (4D transport compression), Charles Han (normal map filtering), Ira Kemelmacher-Shlizerman, Dhruv Mahajan and Yu-Ting Tseng (locally low-dimensional light transport and BRDF factorization), Ryan Overbeck (temporal coherence), Bo Sun (volumetric scattering and affine double products), and faculty colleagues: Peter Belhumeur, Fredo Durand, Eitan Grinspun, and Greg Humphreys. I also thank the anonymous reviewer very much for his/her careful reading of this lengthy manuscript, and the very detailed suggestions made. Finally, I wish to thank the NSF for supporting my work on precomputation-based rendering with two grants (jointly with Henrik Wann Jensen), # 0305322

and currently # 0701775. This work is also supported in part by NSF grants # 0446916, # 0541259, a Sloan Research Fellowship, an ONR Young Investigator Award, and equipment donations from Intel and NVIDIA.

# References


[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. Jensen, "Structured importance sampling of environment maps," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 03*, vol. 22, no. 3, pp. 605–612, 2003.

[2] J. Airey, J. Rohlf, and F. Brooks, "Towards image realism with interactive update rates in complex virtual building environments," *ACM Symposium on Interactive 3D Graphics*, pp. 41–50, 1990.

[3] O. Akerlund, M. Unger, and R. Wang, "Precomputed visibility cuts for interactive relighting with dynamic BRDFs," *Pacific Graphics 07*, 2007.

[4] T. Annen, Z. Dong, T. Mertens, P. Bekaert, and H. Seidel, "Real-time all-frequency shadows in dynamic scenes," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 08*, vol. 27, no. 3, pp. 1–8, (Article 34), 2008.

[5] T. Annen, J. Kautz, F. Durand, and H. Seidel, "Spherical harmonic gradients for mid-range illumination," *EuroGraphics Symposium on Rendering*, pp. 331–336, 2004.

[6] J. Arvo, "Applications of irradiance tensors to the simulation of non-Lambertian phenomena," in *Proceedings of SIGGRAPH 95*, pp. 335–342, 1995.

[7] J. Arvo, K. Torrance, and B. Smits, "A framework for the analysis of error in global illumination algorithms," in *Proceedings of SIGGRAPH 94*, pp. 75–84, 1994.

[8] M. Ashikhmin and P. Shirley, "An anisotropic phong BRDF model," *Journal of Graphics Tools*, vol. 5, no. 2, pp. 25–32, 2000.

[9] M. Ashikhmin and P. Shirley, "Steerable illumination textures," *ACM Transactions on Graphics*, vol. 21, no. 1, pp. 1–19, 2002.

[10] J. Barbic and D. James, "Real-time subspace integration of St. Venant-Kirchoff Deformable Models," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 982–990, 2005.

[11] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *International Conference on Computer Vision*, pp. 383–390, 2001.

[12] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[13] A. Ben-Artzi, K. Egan, R. Ramamoorthi, and F. Durand, "A precomputed polynomial representation for interactive BRDF editing with global illumination," *ACM Transactions on Graphics*, vol. 27, no. 2, pp. 1–13, (Article 13), 2008.

[14] A. Ben-Artzi, R. Overbeck, and R. Ramamoorthi, "Real-time BRDF editing in complex lighting," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 945–954, 2006.

[15] J. Blinn and M. Newell, "Texture and reflection in computer generated images," *Communications of the ACM*, pp. 542–547, October 1976.

[16] D. Blythe, "The Direct3D 10 system," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 724–734, 2006.

[17] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, and P. Hanrahan, "Brook for GPUs: Stream computing on graphics hardware," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 04*, vol. 23, pp. 777–786, August 2004.

[18] B. Cabral, N. Max, and R. Springmeyer, "Bidirectional reflection functions from surface bump maps," in *Proceedings of SIGGRAPH 87*, pp. 273–281, 1987.

[19] B. Cabral, M. Olano, and P. Nemec, "Reflection space image based rendering," in *Proceedings of SIGGRAPH 99*, pp. 165–170, 1999.

[20] W. Chen, J. Bouguet, M. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 02*, vol. 21, no. 3, pp. 447–456, 2002.

[21] E. Cheslack-Postava, N. Goodnight, R. Ng, R. Ramamoorthi, and G. Humphreys, "4D compression and relighting with high-resolution light transport matrices," *ACM Symposium on Interactive 3D Graphics and Games*, pp. 81–88, 2007.

[22] E. Cheslack-Potava, R. Wang, O. Akerlund, and F. Pellacini, "Fast, realistic lighting and material design using nonlinear cut approximation," *ACM Transactions on Graphics, Proceedings of SIGGRAPH Asia 08*, vol. 27, no. 5, pp. 1–10 (Article 128), 2008.

[23] P. Clarberg, W. Jarosz, T. Moller, and H. Jensen, "Wavelet importance sampling: Evaluating products of complex functions," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1166–1175, 2005.

[24] M. Cohen and J. Wallace, *Radiosity and Realistic Image Synthesis*. Academic Press, 1993.

[25] F. Crow, "Summed-area tables for texture mapping," in *Proceedings of SIGGRAPH 84*, pp. 207–212, 1984.

[26] K. Dana, B. Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics*, vol. 18, pp. 1–34, January 1999.

[27] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications of Pure and Applied Mathematics*, vol. 41, pp. 909–996, 1988.

[28] P. Debevec, "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *Proceedings of SIGGRAPH 98*, pp. 189–198, 1998.

[29] P. Debevec, T. Hawkins, C. Tchou, H. P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the reflectance field of a human face," in *Proceedings of SIGGRAPH 00*, pp. 145–156, 2000.

[30] P. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proceedings of SIGGRAPH 97*, pp. 369–378, 1997.

[31] P. Debevec, http://www.debevec.org/ReflectionMapping/.

[32] P. Debevec, http://www.debevec.org/probes/.

[33] R. Devore, "Nonlinear approximation," *Acta Numerica*, vol. 7, pp. 51–150, 1998.

[34] Y. Dobashi, K. Kaneda, H. Nakatani, H. Yamashita, and T. Nishita, "A quick rendering method using basis functions for interactive lighting design," *EuroGraphics 95*, pp. 229–240, 1995.

[35] J. Dorsey, J. Arvo, and D. Greenberg, "Interactive design of complex time-dependent lighting," *IEEE Computer Graphics and Applications*, vol. 15, no. 2, pp. 26–36, 1995.

[36] W. Freeman and T. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[37] P. Gautron, J. Krivanek, S. Pattanaik, and K. Bouatouch, "A novel hemispherical basis for accurate and efficient rendering," *EuroGraphics Symposium on Rendering*, pp. 321–330, 2004.

[38] R. Gershbein and P. Hanrahan, "A fast relighting engine for interactive cinematic lighting design," in *Proceedings of SIGGRAPH 00*, pp. 353–358, 2000.

[39] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[40] G. Golub and C. V. Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.

[41] C. Goral, K. Torrance, D. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," in *Proceedings of SIGGRAPH 84*, pp. 213–222, 1984.

[42] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *Proceedings of SIGGRAPH 96*, pp. 43–54, 1996.

[43] P. Green, J. Kautz, and F. Durand, "Efficient reflectance and visibility approximations for environment map rendering," *Computer Graphics Forum, EuroGraphics 07*, vol. 26, no. 3, pp. 495–502, 2007.

[44] P. Green, J. Kautz, W. Matusik, and F. Durand, "View-dependent precomputed light transport using nonlinear gaussian function approximations," *ACM Symposium on Interactive 3D Graphics and Games*, pp. 7–14, 2006.

[45] R. Green, "Spherical Harmonic Lighting: The Gritty Details," http://www.research.scea.com/gdc-2003/spherical-harmonic-lighting.pdf.

[46] N. Greene, "Environment mapping and other applications of world projections," *IEEE Computer Graphics & Applications*, vol. 6, no. 11, pp. 21–29, 1986.

[47] J. Gu, C. Tu, R. Ramamoorthi, P. Belhumeur, W. Matusik, and S. Nayar, "Time-varying surface appearance: Acquisition, modeling and rendering," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 762–771, 2006.

[48] P. Haeberli and M. Segal, "Texture mapping as a fundamental drawing primitive," *EuroGraphics Workshop on Rendering*, pp. 259–266, 1993.

[49] Z. Hakura and J. Snyder, "Realistic reflections and refractions on graphics hardware with hybrid rendering and layered environment maps," *Eurographics Workshop on Rendering*, pp. 289–300, 2001.

[50] Z. Hakura, J. Snyder, and J. Lengyel, "Parameterized environment maps," *ACM Symposium on Interactive 3D Graphics*, pp. 203–208, 2001.

[51] C. Han, B. Sun, R. Ramamoorthi, and E. Grinspun, "Frequency domain normal map filtering," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 07*, vol. 26, no. 3, pp. 1–11 (Article 28), 2007.

[52] M. Hasan, F. Pellacini, and K. Bala, "Direct to indirect transfer for cinematic relighting," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 1089–1097, 2005.

[53] M. Hasan, F. Pellacini, and K. Bala, "Matrix row-column sampling for the many-light problem," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 07*, vol. 26, no. 3 (Article 26), 2007.

[54] V. Havran and J. Bittner, "LCTS: Ray shooting using longest common traversal sequences," *Computer Graphics Forum*, vol. 19, no. 3, p. 59, 2000.

[55] W. Heidrich, H. Lensch, M. Cohen, and H. Seidel, "Light field techniques for reflections and refractions," *EuroGraphics Workshop on Rendering*, pp. 187–196, 1999.

[56] W. Heidrich and H.-P. Seidel, "Realistic, hardware-accelerated shading and lighting," in *Proceedings of SIGGRAPH 99*, pp. 171–178, August 1999.

[57] B. Hill and P. Faloutsos, "Photorealistic lighting with offset radiance transfer mapping," *ACM Symposium on Interactive 3D Graphics and Games*, pp. 15–21, 2006.

[58] T. Inui, Y. Tanabe, and Y. Onodera, *Group Theory and Its Applications in Physics*. Springer Verlag, 1990.

[59] K. Iwasaki, Y. Dobashi, F. Yoshimoto, and T. Nishita, "Precomputed radiance transfer for dynamic scenes taking into account light interreflection," *EuroGraphics Symposium on Rendering*, pp. 35–44, 2007.

[60] J. Jackson, *Classical Electrodynamics*. John Wiley, 1975.

[61] D. James and K. Fatahalian, "Precomputing interactive dynamic deformable scenes," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 03*, vol. 22, no. 3, pp. 879–887, 2003.

[62] H. Jensen, *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.

[63] H. Jensen, S. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proceedings of SIGGRAPH 01*, pp. 511–518, 2001.

[64] J. Kajiya, "The rendering equation," in *Proceedings of SIGGRAPH 86*, pp. 143–150, 1986.

[65] N. Kambhatla and T. Leen, "Fast non-linear dimension reduction," *Advances in Neural Information Processing Systems 6*, 1994.

[66] N. Kambhatla and T. Leen, "Dimension reduction by local PCA," *Neural Computation*, vol. 9, p. 1493, 1997.

[67] J. Kautz, "Hardware lighting and shading: A survey," *Computer Graphics Forum*, vol. 23, no. 1, pp. 85–112, 2004.

[68] J. Kautz, S. Boulos, and F. Durand, "Interactive editing and modeling of bidirectional texture functions," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 07*, vol. 26, no. 3 (Article 53), 2007.

[69] J. Kautz, J. Lehtinen, and T. Aila, "Hemispherical rasterization for self-shadowing of dynamic objects," *EuroGraphics Symposium on Rendering*, pp. 179–184, 2004.

[70] J. Kautz and M. McCool, "Approximation of glossy reflection with prefiltered environment maps," *Graphics Interface*, pp. 119–126, 2000.

[71] J. Kautz, P. Sloan, and J. Snyder, "Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics," *EuroGraphics Workshop on Rendering*, pp. 301–308, 2002.

[72] J. Kautz, P. Vázquez, W. Heidrich, and H. Seidel, "A unified approach to prefiltered environment maps," *EuroGraphics Rendering Workshop 00*, pp. 185–196, 2000.

[73] A. Keller, "Instant radiosity," in *Proceedings of SIGGRAPH 97*, pp. 49–56, 1997.

[74] J. Koenderink and A. van Doorn, "Phenomenological description of bidirectional surface reflection," *JOSA A*, vol. 15, no. 11, pp. 2903–2912, 1998.

[75] J. Kontkanen and T. Aila, "Ambient occlusion for animated characters," *EuroGraphics Symposium on Rendering 06*, pp. 343–348, 2006.

[76] J. Kontkanen and S. Laine, "Ambient occlusion fields," *ACM Symposium on Interactive 3D Graphics and Games*, pp. 41–48, 2005.

[77] J. Kontkanen, E. Turquin, N. Holzschuch, and F. Sillion, "Wavelet radiance transport for real-time indirect lighting," *EuroGraphics Symposium on Rendering 06*, pp. 161–172, 2006.

[78] A. Kristensen, T. Moller, and H. Jensen, "Precomputed local radiance transfer for real-time lighting design," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1208–1215, 2005.

[79] J. Lehtinen, "A framework for precomputed and captured light transport," *ACM Transactions on Graphics*, vol. 26, no. 4, pp. 1–22 (Article 13), 2007.

[80] J. Lehtinen and J. Kautz, "Matrix radiance transfer," *ACM Symposium on Interactive 3D Graphics*, pp. 59–64, 2003.

[81] J. Lehtinen, M. Zwicker, E. Turquin, J. Kontkanen, F. Durand, F. Sillion, and T. Aila, "A meshless hierarchical representation for light transport," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 08*, vol. 27, no. 3, pp. 1–9 (Article 37), 2008.

[82] C. Lessig and E. Fiume, "SOHO: Orthogonal and symmetric haar wavelets on the sphere," *ACM Transactions on Graphics*, vol. 27, no. 1, pp. 1–11 (Article 4), 2008.

[83] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of SIGGRAPH 96*, pp. 31–42, 1996.

[84] E. Lindholm, M. J. Kilgard, and H. Moreton, "A user-programmable vertex engine," in *Proceedings of SIGGRAPH 01*, pp. 149–158, August 2001.

[85] X. Liu, P. Sloan, H. Shum, and J. Snyder, "All-frequency precomputed radiance transfer for glossy objects," *EuroGraphics Symposium on Rendering 04*, pp. 337–344, 2004.

[86] T. MacRobert, *Spherical Harmonics: An Elementary Treatise on Harmonic Functions with Applications*. Dover Publications, 1948.

[87] D. Mahajan, I. Kemelmacher-Shlizerman, R. Ramamoorthi, and P. Belhumeur, "A theory of locally low dimensional light transport," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 07*, vol. 26, no. 3, p. 62, 2007.

[88] D. Mahajan, Y. Tseng, and R. Ramamoorthi, "An analysis of the in-out BRDF factorization for view-dependent relighting," *Computer Graphics Forum, EGSR 08*, vol. 27, no. 4, pp. 1137–1145, 2008.

[89] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[90] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard, "Cg: A system for programming graphics hardware in a C-like language," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 03*, vol. 22, no. 3, pp. 896–907, July 2003.

[91] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan, "Image-based 3D photography using opacity hulls," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 02*, vol. 21, no. 3, pp. 427–437, 2002.

[92] C. Mei, J. Shi, and F. Wu, "Rendering with spherical radiance transport maps," *Computer Graphics Forum, Eurographics 04*, vol. 23, no. 3, pp. 281–290, 2004.

[93] G. Miller and C. Hoffman, "Illumination and reflection maps: Simulated objects in simulated and real environments," *SIGGRAPH 84 Advanced Computer Graphics Animation Seminar Notes*, http://www.debevec.org/ReflectionMapping/illum.pdf, 1984.

[94] G. Miller, S. Rubin, and D. Poncelon, "Lazy decompression of surface light fields for precomputed global illumination," *Eurographics Rendering Workshop 98*, pp. 281–292, 1998.

[95] S. Nayar, P. Belhumeur, and T. Boult, "Lighting-sensitive displays," *ACM Transactions on Graphics*, vol. 23, no. 4, pp. 963–979, 2004.

[96] R. Ng, R. Ramamoorthi, and P. Hanrahan, "All-frequency shadows using non-linear wavelet lighting approximation," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 03*, vol. 22, no. 3, pp. 376–381, 2003.

[97] R. Ng, R. Ramamoorthi, and P. Hanrahan, "Triple product wavelet integrals for all-frequency relighting," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 04*, vol. 23, no. 3, pp. 475–485, 2004.

[98] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometric Considerations and Nomenclature for Reflectance, NBS Monograph 160*. US, National Bureau of Standards, 1977.

[99] J. Nimeroff, E. Simoncelli, and J. Dorsey, "Efficient re-rendering of naturally illuminated environments," *EuroGraphics Workshop on Rendering*, pp. 359–373, 1994.

[100] K. Nishino, S. Nayar, and T. Jebara, "Clustered blockwise PCA for representing visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1675–1679, 2005.

[101] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-texture method: Appearance compression and synthesis based on a 3D model," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), vol. 23, no. 11, pp. 1257–1265, 2001.

[102] R. Overbeck, A. Ben-Artzi, R. Ramamoorthi, and E. Grinspun, "Exploiting temporal coherence for incremental all-frequency relighting," *EuroGraphics Symposium on Rendering*, pp. 151–160, 2006.

[103] P. Peers, D. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec, "Compressive light transport sensing," *ACM Transactions on Graphics*, vol. 28, no. 1, pp. 1–18 (Article 3), 2009.

[104] F. Pellacini, K. Vidimce, A. Lefohn, A. Mohr, M. Leone, and J. Warren, "Lpics: A hybrid hardware-accelerated relighting engine for computer cinematography," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 464–470, 2005.

[105] K. Proudfoot, W. R. Mark, S. Tzvetkov, and P. Hanrahan, "A real-time procedural shading system for programmable graphics hardware," in *Proceedings of SIGGRAPH 01*, pp. 159–170, August 2001.

[106] T. Purcell, I. Buck, W. Mark, and P. Hanrahan, "Ray tracing on programmable graphics hardware," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 02*, vol. 21, pp. 703–712, July 2002.

[107] J. Ragan-Kelley, C. Kilpatrick, B. Smith, D. Epps, P. Green, C. Hery, and F. Durand, "The Lightspeed automatic interactive lighting preview system," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 07*, vol. 26, no. 3 (Article 25), 2007.

[108] R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proceedings of SIGGRAPH 01*, pp. 497–500, 2001.

[109] R. Ramamoorthi and P. Hanrahan, "On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object," *Journal of the Optical Society of America A*, vol. 18, no. 10, pp. 2448–2459, 2001.

[110] R. Ramamoorthi and P. Hanrahan, "A signal-processing framework for inverse rendering," in *Proceedings of SIGGRAPH 01*, pp. 117–128, 2001.

[111] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 02*, vol. 21, no. 3, pp. 517–526, 2002.

[112] R. Ramamoorthi and P. Hanrahan, "A signal-processing framework for reflection," *ACM Transactions on Graphics*, vol. 23, no. 4, pp. 1004–1042, 2004.

[113] Z. Ren, R. Wang, J. Snyder, K. Zhou, X. Liu, B. Sun, P. Sloan, H. Bao, Q. Peng, and B. Guo, "Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 977–986, 2006.

[114] A. Reshetov, A. Soupikov, and J. Hurley, "Multi-level ray tracing algorithm," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1176–1185, 2005.

[115] S. Rusinkiewicz, "A new change of variables for efficient BRDF representation," *EuroGraphics Workshop on Rendering*, pp. 11–22, 1998.

[116] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," in *Proceedings of SIGGRAPH 95*, pp. 161–172, 1995.

[117] F. Sillion, J. Arvo, S. Westin, and D. Greenberg, "A global illumination solution for general reflectance distributions," in *Proceedings of SIGGRAPH 91*, pp. 187–196, 1991.

[118] P. Sloan, "Stupid spherical harmonics tricks," http://www.ppsloan.org/publications/StupidSH36.pdf.

[119] P. Sloan, "Normal mapping for precomputed radiance transfer," *ACM Symposium on Interactive 3D Graphics and Games*, 2006.

[120] P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered principal components for precomputed radiance transfer," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 03*, vol. 22, no. 3, pp. 382–391, 2002.

[121] P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 02*, vol. 21, no. 3, pp. 527–536, 2002.

[122] P. Sloan, X. Liu, H. Shum, and J. Snyder, "Bi-Scale radiance transfer," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 03*, vol. 22, no. 3, pp. 370–375, 2003.

[123] P. Sloan, B. Luna, and J. Snyder, "Local, deformable precomputed radiance transfer," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1216–1224, 2005.

[124] E. Stollnitz, T. DeRose, and D. Salesin, *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, 1996.

[125] B. Sun and R. Ramamoorthi, "Affine double and triple product wavelet integrals for rendering," *ACM Transaction on Graphics*, vol. 28, no. 2, 2009.

[126] B. Sun, R. Ramamoorthi, S. Narasimhan, and S. Nayar, "A practical analytic single scattering model for real-time rendering," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1040–1049, 2005.

[127] B. Sun, K. Sunkavalli, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Time-varying BRDFs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 595–609, 2007.

[128] W. Sun and A. Mukherjee, "Generalized wavelet product integral for rendering dynamic glossy objects," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 955–966, 2006.

[129] X. Sun, K. Zhou, Y. Chen, S. Lin, J. Shi, and B. Guo, "Interactive relighting with dynamic BRDFs," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 07*, vol. 26, no. 3, pp. 1–10, (Article 27), 2007.

[130] X. Sun, K. Zhou, E. Stollnitz, J. Shi, and B. Guo, "Interactive relighting of dynamic refractive objects," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 08*, vol. 27, no. 3, pp. 1–9 (Article 35), 2008.

[131] S. Teller and J. Alex, "Frustum casting for progressive, interactive rendering," Technical Report, MIT/LCS/TR-740, MIT, 1998.

[132] P. Teo, E. Simoncelli, and D. Heeger, "Efficient linear re-rendering for interactive lighting design," Technical Report, STAN-CS-TN-97-60, Stanford, 1997.

[133] Y. Tsai and Z. Shih, "All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 967–976, 2006.

[134] Y. J. W. Feng, L. Peng and Y. Yu, "Large-scale data management for PRT-based real-time rendering of dynamically skinned models," *EuroGraphics Symposium on Rendering*, 2007.

[135] I. Wald, T. Ize, A. Kensler, A. Knoll, and S. G. Parker, "Ray tracing animated scenes using coherent grid traversal," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 06*, vol. 25, no. 3, pp. 485–493, 2006.

[136] I. Wald, W. R. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. G. Parker, and P. Shirley, "State of the art in ray tracing animated scenes," in *STAR Proceedings of Eurographics 07*, (D. Schmalstieg and J. Bittner, eds.), pp. 89–116, The Eurographics Association, September 2007.

[137] I. Wald, P. Slusallek, C. Benthin, and M. Wagner, "Interactive rendering with coherent ray tracing," *Computer Graphics Forum*, vol. 20, no. 3, pp. 153–164, 2001.

[138] J. Wang, X. Tong, S. Lin, M. Pan, C. Wang, H. Bao, B. Guo, and H. Shum, "Appearance manifolds for modeling time-variant appearance of materials," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 754–761, 2006.

[139] J. Wang, K. Xu, K. Zhou, S. Lin, S. Hu, and B. Guo, "Spherical harmonics scaling," *The Visual Computer, Pacific Graphics 06*, vol. 22, no. 9–11, pp. 713–720, 2006.

[140] L. Wang, W. Wang, J. Dorsey, X. Yang, B. Guo, and H. Shum, "Real-time rendering of plant leaves," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 712–719, 2005.

[141] R. Wang, E. Cheslack-Potava, R. Wang, D. Luebke, Q. Chen, W. Hua, Q. Peng, and H. Bao, "Real-time editing and relighting of homogeneous translucent materials," *The Visual Computer, CGI 08*, vol. 24, no. 7–9, pp. 565–575, 2008.

[142] R. Wang, R. Ng, D. Luebke, and G. Humphreys, "Efficient wavelet rotation for environment map rendering," *EuroGraphics Symposium on Rendering*, pp. 173–182, 2006.

[143] R. Wang, J. Tran, and D. Luebke, "All-frequency relighting of non-diffuse objects using separable BRDF approximation," *EuroGraphics Symposium on Rendering*, pp. 345–354, 2004.

[144] R. Wang, J. Tran, and D. Luebke, "All-frequency interactive relighting of translucent objects with single and multiple scattering," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1202–1207, 2005.

[145] R. Wang, J. Tran, and D. Luebke, "All-frequency relighting of glossy objects," *ACM Transactions on Graphics*, vol. 25, no. 2, pp. 293–318, 2006.

[146] R. Wang, J. Zhu, and G. Humphreys, "Precomputed radiance transfer for real-time indirect lighting using a spectral mesh basis," *EuroGraphics Symposium on Rendering*, 2007.

[147] T. Whitted, "An improved illumination model for shaded display," *Communications of the ACM*, vol. 23, no. 6, pp. 343–349, 1980.

[148] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle, "Surface light fields for 3D photography," in *Proceedings of SIGGRAPH 00*, pp. 287–296, 2000.

[149] S. Woop, J. Schmittler, and P. Slusallek, "RPU: a programmable ray processing unit for realtime ray tracing," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, pp. 434–444, August 2005.

[150] K. Zhou, Y. Hu, S. Lin, B. Guo, and H. Shum, "Precomputed shadow fields for dynamic scenes," *ACM Transactions on Graphics, Proceedings of SIGGRAPH 05*, vol. 24, no. 3, pp. 1196–1201, 2005.