

Contents

1	Introduction	1
1.1	Realistic Image Synthesis	1
1.2	Contributions	3
1.3	Organization of This Thesis	3
2	Fundamentals of Realistic Image Synthesis	5
2.1	Preliminaries	5
2.2	Physically Based Model of Light	7
2.3	Interaction of Light with Surfaces	10
2.4	Mathematical Principles of Realistic Image Synthesis	12
2.4.1	The Rendering Equation	12
2.4.2	Vector Spaces	13
2.4.3	Linear Operators	16
2.4.4	The Rendering Equation Revisited	17
3	Previous Work	19
3.1	Approximation of Environment Illumination	19
3.2	Precomputed Transfer	22
3.3	Real-Time Shadow Techniques	27
4	A Hybrid Algorithm for Environment Illumination Rendering	29
4.1	Overview of the Algorithm	29
4.2	Precomputation Phase	30
4.3	Decomposition of the Environment Illumination	31

4.3.1	Decomposition Model	33
4.3.2	Strategies for Constructing the Approximation	34
4.3.3	Decomposition Algorithm	35
4.4	Rendering	40
5	Experimental Results	43
5.1	Test Setup	43
5.1.1	Test Scenes	43
5.2	Results	45
5.2.1	Visual Comparison	45
5.2.2	Quantitative Comparison	46
5.2.3	Additional Results	49
5.3	Strengths and Weaknesses of the Hybrid Rendering Algorithm	50
5.3.1	Strengths of the Algorithm	51
5.3.2	Weaknesses of the Algorithm	55
6	Discussion and Future Work	59
6.1	Discussion	59
6.2	Conclusions and Future Work	59
	Bibliography	61

Chapter 1

Introduction

1.1 Realistic Image Synthesis

Visual realism has been one of the driving forces in computer graphics over the last few decades. Considerable research efforts have been devoted to capture the appearance of real-world lighting in computer generated imagery. By realistic image synthesis, we mean the branch of computer graphics research aiming to generate photorealistic images by means of physically based computer simulation. Traditionally the simulations have been computationally demanding, effectively prohibiting their use in a real-time setting, such as computer games or architectural walkthroughs. In this thesis, we aim to increase the quality of real-time generated images by effectively combining approximative real-time rendering techniques with physically based light transport simulation.

The importance of realistic image synthesis can be measured in terms of its applications. Realistic image synthesis has enabled film and visual effects professionals to bypass some technological obstacles limiting their creative potential. Entire feature-length films are being produced with the aid of computers. Computer and video games are beginning to take advantage of *global illumination* effects, such as smooth shadows, in order to create better gaming experiences. The flourishing video game industry has on its own part accelerated the development of new rendering algorithms and dedicated graphics hardware. Besides the obvious applications in the entertainment industry, realistic image synthesis has been successfully applied in simulators and design visualization. Architectural design benefits from the ability to visualize the indoor illumination due to interior lighting as well as to render images under varying outdoor illumination. Realistic image synthesis is ubiquitous in the advertising world. Future product concepts are easily promoted with the aid of computer generated images.

Photorealistic pictures are generated by simulating the behavior of light in the given scenes. The simulation resembles the real physical process governing the light transport between physical objects. One usually need not consider interactions in the small, i.e., phenomena described fully by the quantum theory of physics. Instead, the simulation is usually based on geometric optics and takes place in a vacuum with no participating media. The general problem of realistic image synthesis can then be formulated as follows: given

- a set of points \mathcal{G} in \mathbb{R}^3 representing the geometrical surfaces of the scene. The set \mathcal{G} obviously needs to have additional structure so that common geometrical operations and objects are well defined. We will consider this augmented structure in more detail in Chapter 2. For now, it

will suffice to imagine a typical computer graphics scene and identify the visible geometry with the set \mathcal{G} .

- a set of materials \mathcal{M} and an assignment of materials to surface points $\Theta : \mathcal{G} \rightarrow \mathcal{M}$,
- a set of light sources \mathcal{E} ,

compute the steady state distribution of light emitted from the light sources \mathcal{E} on the geometry \mathcal{G} simulating the behavior of light taking into account the inherent effects of the different materials \mathcal{M} to the resulting distribution.

The geometry \mathcal{G} of the scene will often consist of geometric primitives such as triangles, quadrangles or spline surfaces. The geometry can also be described procedurally with a strict language and a set of transformation rules for the language, e.g., L-systems.

The material assignment function Θ and the set of materials \mathcal{M} uniquely characterize the scene under different lighting configurations. A material $m \in \mathcal{M}$ approximates real-world light-object interaction mechanisms that transform incident light distributions to reflected and transmitted light distributions. Examples of such phenomena are reflection, transmission and absorption.

The set of light sources \mathcal{E} includes information about the position, orientation, geometric shape and radiance distribution on the light source geometry for each light $l \in \mathcal{E}$. Together with the surfaces \mathcal{G} and the materials \mathcal{M} , the set of lights \mathcal{E} uniquely characterizes the scene in terms of light transport.

Realistic images synthesis is sometimes referred to as *global illumination* and algorithms that solve the realistic image synthesis problem are correspondingly called global illumination algorithms. This is due to the global nature of the problem, the steady state light distribution at given surface point x might potentially depend on all the other surface points in a nontrivial way. This is in contrast to *local illumination*. Local illumination uses only local information, such as surface normal vectors to determine the light distribution for each surface point.

The strive towards realism has benefited from the advances in computer hardware. In addition to development of PC-hardware, a new type of specialized hardware, known as *Graphics Processing Units* or *GPUs* have become commonplace as well. Their highly-parallel structure makes them more effective than typical CPUs for common graphics operations, such as *vertex processing* and *rasterization*. Recent GPUs have included support for programmability. Programmability has greatly enlarged the set of algorithms that could be implemented on a GPU. It has also affected the design of new algorithms to include support for a GPU implementation. In addition to the more general GPUs, there has also been efforts to implement specific global illumination algorithms totally in hardware [61].

Hardware development has directed the research efforts as well. Recently, there has been a shift of focus in research from offline rendering to real-time rendering. Real-time methods usually employ a wide and unrestricted selection of approximations in order to limit the computational complexity of the image synthesis problem. The goal of research in real-time computer graphics is to get rid of as many as possible approximations while increasing the image quality to parallel that of offline algorithms.

This thesis positions itself halfway between the real-time and the offline rendering realms, with an emphasis on the real-time realm. We propose novel methods in order to increase the visual realism of real-time computer graphics by taking advantage of state-of-the-art methods in both real-time and offline rendering.

More specifically, we consider the rendering of static scenes under distant, time-varying *environment illumination* in real-time. Environment illumination can be visualized by imagining how a small

1.2 Contributions

reflective sphere would look like in the scene of interest, see Figure 5.1 in Chapter 5. We limit our consideration to *diffuse* surfaces, i.e., surfaces which reflect incident light uniformly over all directions.

1.2 Contributions

We propose a novel hybrid rendering algorithm for realistic rendering of static scenes under distant environment illumination. Our contributions are twofold. Firstly, we combine real-time shadowing techniques with precomputation-based techniques. Secondly, we introduce a decomposition method for *spherical functions*. By spherical functions we mean real valued functions defined on the sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$. It should be noted that distant environment illumination can be identified with a spherical function. In the rest of this thesis we identify the term environment illumination with the spherical function representing the illumination. Even though we focus our study on real valued spherical functions, it is straightforward to generalize the concept to vector valued mappings to account for wavelength variation in the environment illumination simply by constructing the vector valued mapping from several real valued component functions.

We approximate the environment illumination using a combination of linear basis projection and *multilobe* approximation methods [66, 80, 44, 37]. At runtime, we render the scene using directional light sources and shadow mapping to approximate the lighting contribution of the multilobe component together with a *precomputed radiance transfer* (PRT) solution to approximate the lighting by the linear basis functions. In comparison to previous methods [66, 53], we are able to approximate both sharp and blurry shadows originating from detailed environment illumination, with small additional runtime cost.

It is well known that real-time rendering techniques are not well equipped to handle smooth shadows nor other global illumination effects caused by area light sources [66]. On the other hand, previous precomputed transfer methods have not succeeded in reproducing sharp shadows caused by concentrated environmental light sources, unless they revert into massive precomputation which leads to hundreds of megabytes of precomputed data [53].

Our proposed method combines the strengths of the previous methods, while avoiding their most significant pitfalls, resulting in an approximative real-time rendering algorithm for static scenes. Namely, we use a diffuse PRT solution to capture the blurry shadows caused by smoothly varying environment illumination together with shadow mapping to reproduce sharp shadows caused by concentrated light sources.

1.3 Organization of This Thesis

The rest of this thesis is organized as follows. The fundamentals of realistic image synthesis are reviewed in Chapter 2. In Chapter 3 we examine previous work related to environment illumination, precomputed light transport and real-time methods for approximating the solutions to the realistic image synthesis problem. In Chapter 4 we introduce our novel algorithm for real-time rendering of static scenes under time-varying, distant environment illumination. We study the properties of real-world environment illumination and introduce the concept of environment illumination decomposition. We also discuss various approaches that can be used to analyze environment illumination and to obtain the desired decomposition. We also consider some practical issues related to the implementation of the algorithm. Experimental results of the algorithm are presented in Chapter 5.

Finally, we conclude this thesis by giving some directions for future work in Chapter 6.

Chapter 2

Fundamentals of Realistic Image Synthesis

This chapter provides an overview of the principles of realistic image synthesis by reviewing some of the physics behind light transport. The theory of realistic image synthesis aims in formulating an abstraction to the physical processes that govern light transport. The resulting theory is based on a mixture of different but interrelated sciences, such as psychology, digital signal processing, mathematics and physics.

We use the theory of realistic image synthesis as a basis for the development of the proposed rendering method. Therefore we claim that the proposed rendering method falls under the category of realistic image synthesis. Another reason for studying the theory of realistic image synthesis is to understand and analyze the sources of approximation error in the final result, i.e., the synthesized image.

We begin by introducing some preliminary terms and definitions and then we consider radiometry as a means for quantifying the light transport process. When we have the radiometric concepts at hand, we study the interaction of light with surfaces. Finally, we introduce the rendering equation, which is a restatement of the realistic image synthesis problem as a concrete mathematical object, that is, as an integral equation. We conclude this chapter by introducing a natural mathematical framework for manipulating the equations of global illumination.

2.1 Preliminaries

In this section we list some basic terms and definitions used throughout the thesis.

Surface. We assume, that the scene geometry is given as a collection of points $\mathcal{G} \in \mathbb{R}^3$, representing the geometrical surfaces of the scene. We denote the members of the set \mathcal{G} as *surface points*. In addition, we assume that for all $x \in \mathcal{G}$ we are given

- well-defined surface normals $N(x) \in \mathbb{S}^2$,
- well-defined tangent planes $\Gamma(x) \subset \mathbb{R}^3$. The surface normal vector $N(x)$ is orthogonal to each tangent vector $T \in \Gamma(x)$ with respect to the natural inner product of \mathbb{R}^3 , i.e., $T \cdot N(x) = 0$. In

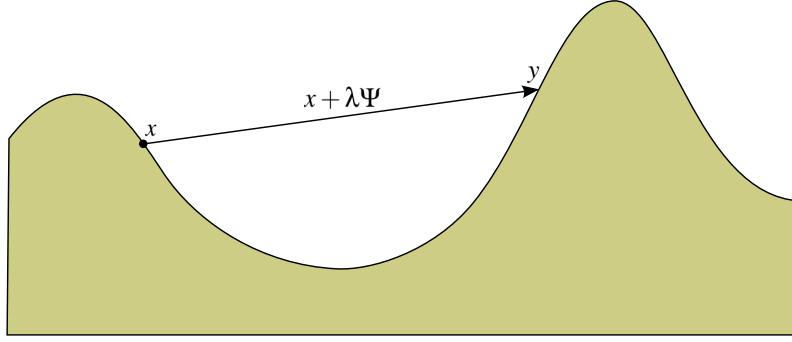


Figure 2.1 Illustration of the raycast operation $R(x, \Psi) = x + \lambda\Psi = y$.

addition the tangent plane $\Gamma(x)$ contains the point x .

The alert reader may notice that certain mathematical objects called (smooth) Euclidean 2-manifolds satisfy the above stated requirements. There would be more to say on this interesting subject but for the purposes of this thesis the above mentioned definition is more than sufficient.

Ray-cast. The ray-cast operation is very intuitive; given a surface point x and a direction ψ the ray-cast operation $R(x, \psi)$ returns a surface point y such that

$$(2.1) \quad \|x - y\| = \inf\{\|x - z\| : z = x + \lambda\psi\},$$

for some positive λ . The ray-cast operation is undefined if the ray $x + \lambda\psi$ does not intersect the model.

It is worthwhile to note, that we do not require any degree of smoothness from the surfaces. Naturally we require that each surface is compatible with the ray-cast operation. Examples of surfaces $s \in \mathcal{G}$ are triangle meshes and spline surfaces.

Visibility. The concept of visibility is of fundamental importance in realistic image synthesis. We define visibility indicator between two points $x, y \in \mathcal{G}$ as follows

$$(2.2) \quad V(x, y) = \begin{cases} 1, & \text{if } N(x) \cdot N(y) > 0 \text{ and} \\ & y = x + \lambda(y - x) \text{ where} \\ & \lambda = \inf\{\lambda' > 0 : \exists y' \in \mathcal{S} : y' = x + \lambda'(y' - x)\} \\ 0, & \text{otherwise} \end{cases}.$$

In other words, two surface points x, y are mutually visible if there exists a line segment connecting the points of minimum length and the intersection of the line segment and the set \mathcal{G} is the set $\{x, y\}$.

Hemisphere. While considering some geometric configuration at a surface location $x \in \mathbb{R}^3$ with a normal vector $N(x)$ perpendicular to the tangent plane $T(x)$ at x , we are often interested in the set of directions spanned by the hemisphere

$$\Omega(x, N(x)) := \{y \in \mathbb{R}^3 : y \cdot N(x) \geq 0 \wedge \|x - y\| = 1\} \subset \mathbb{S}^2,$$

where $\mathbb{S}^2 := \{x \in \mathbb{R}^3 : \|x\| = 1\}$ is the 2-sphere. From the definition of $\Omega(x, N(x))$, it is clear that the hemisphere is oriented such that the origin coincides with the point x and the zenith coincides with the point $x + N$. For each direction $d \in \Omega(x, N(x))$ we can associate an angle θ such that $\cos \theta = d \cdot N(x)$. We often use the shorthand $\Omega(x)$ when we mean $\Omega(x, N(x))$.

2.2 Physically Based Model of Light

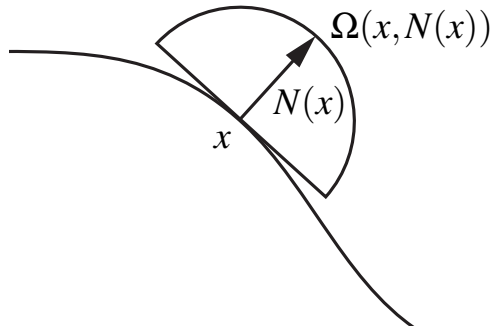


Figure 2.2 Illustration of the hemisphere of points $\Omega(x, N(x))$.

Solid angle. If x is a surface point and Ω' is a subset of $\Omega(x)$, then by the solid angle of Ω' we mean spherical area subtended by the set Ω'^{-1} , i.e., the area on the sphere covered by the set Ω' . The unit of the solid angle is called *steradian*. For instance, the solid angle of the unit 2-sphere is 4π and the solid angle subtended by Europe is roughly $4\pi \frac{10176246}{510065600} \approx 0.25$ steradians, so in total Europe covers about two per cent of the Earth.

2.2 Physically Based Model of Light

In order to formalize the problem of realistic image synthesis we need to establish a sound definition of the intuitive concept of *light*. *Radiometry* provides us with an adequate set of concepts for quantitative analysis of physical light transport. By definition, radiometry is the field of physics that studies the measurement of electromagnetic radiation, including visible light. A related field of study is that of *photometry*, which deals with the quantification of the perception of light energy. Their difference is that photometric quantities are always relative to the brightness perceived by a human eye, whereas radiometric quantities form the absolute, or objective measure of light energy.

We begin our discussion of radiometry by defining the relevant radiometric quantities. Intuitively, light energy can be thought of as a continuous stream of photons, varying in density and wavelength in space over time. There is no point in considering individual photons as, in the context of some measurement apparatus in a continuum, a photon traveling in the continuous stream of photons plays the same role as a real number in a real interval in the context of integration, i.e., insignificant from the perspective of the measurement apparatus.

Thus, from now on we focus our attention to the limit behavior of the photon density in various circumstances. We begin by defining some basic radiometric terms.

Radiant Energy Q . Radiant energy is the energy of electromagnetic waves. Since we are dealing with light energy, we limit our discussion to the visible portion of the electromagnetic spectrum. Intuitively light can be thought of as a natural mechanism of transporting energy through space. Light is a form of energy and therefore it obeys the principle of energy conservation which states that energy is neither created or destroyed in any isolated system. The physical unit of radiant energy is *Joule J*. For example, a light bulb is capable of transforming thermal energy to light energy and a

¹The solid angle subtended by the measurable set Ω' could be more rigorously defined as the Lebesgue measure of the set Ω' .

light-emitting diode (LED) is capable of transforming electric energy to light energy.

Radiant Power Φ . Radiant power is a measure of the total radiant energy per unit time. We define radiant power by

$$(2.3) \quad \Phi := \frac{dQ}{dt}.$$

Radiant power measure is not dependent of the direction of the energy flow so radiant power can express both emitted and received energy per time. The physical unit of radiant power Φ is *Watt* (W).² For example, a typical light source might emit 40 Watt radiant power. We note that the radiant power itself is dependent neither on the geometric properties nor the geometric configuration of the surfaces involved in the radiant energy transport.

Next we introduce two radiometric quantities which relate radiant power to surface geometry and give meaningful interpretation for the direction of radiant energy flow.

Incident Radiosity B_{in} . Incident radiosity is a measure of the incident radiant power per unit surface area

$$(2.4) \quad B_{in} := \frac{d\Phi}{dA} = \frac{d^2Q}{dAdt}.$$

Equation 2.4 says that incident radiosity can be considered as the limit radiant energy perpendicular to a differential surface patch. If $B(x, R)$ is a small neighborhood of the surface point x then incident radiosity is the limit radiant energy as the vector (t, R) approaches zero. The unit of irradiance is therefore W/m^2 . We note that incident radiosity is implicitly dependent of the direction of the radiant energy flow, as it depends explicitly on the differential surface patch. For example, consider a point light source with 40 Watt radiant power which radiates uniformly over all directions. Then the incident radiosity due to the point light source measured at distance $2m$ from the light is $\frac{40}{2^2} W/m^2 = 10W/m^2$ since the area of the incident surface increases with the square of the distance to the light source.

Exitant Radiosity B_{out} . Exitant radiosity $B_{out}(x)$ is a measure of the emitted and reflected radiant power per unit surface area

$$(2.5) \quad B_{out} := \frac{d\Phi}{dA}.$$

Exitant and incident radiosity differ only in the direction of the radiant energy flow. Both quantities tie radiant power with surface geometry via differential surface patches. As an example, a uniform light source with total area of $0.5m^2$ with radiant power of $60W$ has a constant exitant radiosity of $120W/m^2$.

Next we are going to introduce a radiometric quantity which relates radiant power, differential surface patches and differential solid angles.

Incident Radiance $L_{in}(x, \omega)$: $\mathcal{G} \times \mathbb{S}^2 \rightarrow \mathbb{R}$. Incident radiance is a measure of incident radiant power per unit surface area per unit solid angle (Figure 2.3)

$$(2.6) \quad L_{in} := \frac{d^2\Phi}{dA^\perp d\omega} = \frac{d^2\Phi}{\cos\theta dA d\omega}.$$

Equation 2.6 says that incident radiance $L_{in}(x, \omega)$ can be expressed as the limit of radiant energy flowing through the differential solid angle perpendicular to a differential surface patch, hence the

²Watt is a fundamental unit of power, defined by the number of joules per second (*Joule/sec*).

2.2 Physically Based Model of Light

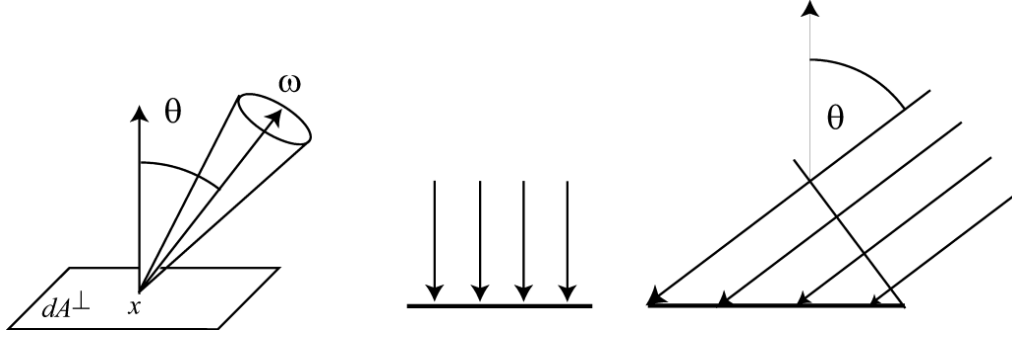


Figure 2.3 (Left) Incident Radiance $L_{\text{in}}(x, \omega)$. (Middle) Incident Radiosity B_{in} . (Right) Directional radiant energy flow [22].

orthogonal projection and the extra cosine term. The limit is taken with respect to time, solid angle and surface area, i.e., radiance is the limit radiant energy as the vector (t, R_1, R_2) approaches zero, where $B(\omega, R_1)$ is a small neighborhood of the direction ω and $B(x, R_2)$ is a small neighborhood of the surface point x and t represents time. We call the incident radiance function L_{in} *field radiance*.

Exitant Radiance $L_{\text{out}}(x, \omega) : \mathcal{G} \times \mathbb{S}^2 \rightarrow \mathbb{R}$. Exitant radiosity is a measure of reflected and emitted radiant power per unit surface area per unit solid angle. Analogously to incident and exitant radiosity, incident and exitant radiance differ only in the direction of the radiant energy flow. We call the exitant radiance function L_{out} *surface radiance*.

Exitant radiance is the most important concept we have defined so far, for the radiance function $L_{\text{out}}(x, \omega)$, can be interpreted as the steady-state distribution of light energy in the scene. Sensors such as digital cameras or the human eye are directly sensitive to radiance. Thus, given the surface radiance L_{out} we could solve the realistic image synthesis problem by measuring the radiance towards the virtual camera from any view location in the scene. In practice it is sufficient to take a finite amount of measurements, say, one for each pixel in the final image. These measurements $M(i, j), i, j = 1, 2, \dots, n$ can be stated in the language of mathematics as follows

$$(2.7) \quad M(i, j) := \int_{A(i, j)} \int_{\Omega(V)} L_{\text{out}}(R(x(i, j), \omega), -\omega) \cos \theta d\omega dA,$$

where (i, j) denotes the pixel in row i column j , $A(i, j)$ denotes the image plane points covered by pixel (i, j) , $x(i, j)$ denotes the world space position of the pixel (i, j) , $\Omega(V)$ denotes the hemispherical set of directions along the view direction V and R is the ray-cast operator.

Relationships between radiometric quantities. By simple calculation, one finds that for diffuse surfaces the exitant radiosity B_{out} is directly proportional to exitant radiance L_{out}

$$(2.8) \quad \pi L_{\text{out}} = B_{\text{out}}.$$

Thus, continuing in the same vein, we can derive the following relationships between the different

Radiometric concept	Area	Orientation	Symbol	Unit
Radiant power	NO	NO	Φ	W
Incident radiosity	YES	NO	B_{in}	$\frac{W}{m^2}$
Exitant radiosity	YES	NO	B_{out}	$\frac{W}{m^2}$
Incident radiance	YES	YES	L_{in}	$\frac{W}{m^2 \text{ steradian}}$
Exitant radiance	YES	YES	L_{out}	$\frac{W}{m^2 \text{ steradian}}$

Table 2.1 A summary of the properties of the radiometric quantities. Area and orientation refer to the connection of the radiometric quantity to geometry.

radiometric quantities.

$$(2.9) \quad \Phi = \int_A \int_{\Omega} L_{in}(x, \omega) \cos \theta d\omega dx$$

$$(2.10) \quad B_{in}(x) = \int_{\Omega} L_{in}(\omega) \cos \theta d\omega$$

$$(2.11) \quad B_{out}(x) = \int_{\Omega} L_{out}(\omega) \cos \theta d\omega$$

The properties of the different radiometric quantities are summarized in Table 2.1.

Each of the radiometric quantities introduced above is implicitly dependent of the wavelength λ . Visible light is usually considered to have wavelength in the range $370nm - 730nm$ [57]. In principle, this means that the light energy distribution should be computed for a number of different wavelengths. This issue has been studied in detail by Roy Hall [31]. In practice, it is usually enough to compute the radiance for only three wavelengths, typically corresponding to the colors of red, green and blue.

Now that we have introduced the most essential concepts in quantifying the physical process that governs light transport and drawn some important connections between the concepts, we are ready to begin the study of the mechanisms that govern the interaction between light and surfaces.

2.3 Interaction of Light with Surfaces

In order to simulate the physically based behavior of light in scenes, we need to establish a set of rules for the transformation mechanisms that alter light paths at surface boundaries. It is well known that light has a dual nature, which is explained by quantum physics. Quantum physics is a fundamental theory that provides accurate and precise descriptions for many phenomena of electromagnetism on the atomic and subatomic level. This model is often considered to be more detailed than is necessary for generating photorealistic images. Instead, computer graphics is usually based on the geometric optics model of light. This model corresponds to our intuitive notion of the behavior of light. The transformation mechanisms in geometric optics state that light can be

- emitted,

2.3 Interaction of Light with Surfaces

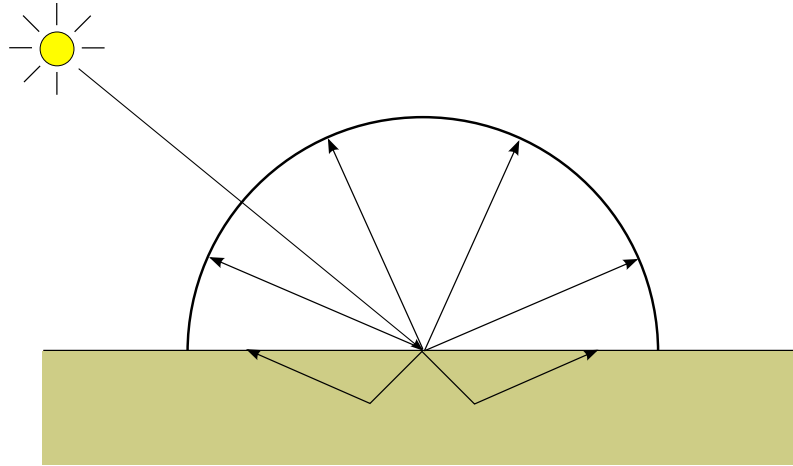


Figure 2.4 Illustration of subsurface scattering.

- reflected,
- transmitted,
- and absorbed

Several approximations are made regarding the behavior of light. In particular, the geometric optics model assumes that light travels instantaneously through a medium in straight lines. In this thesis we do not consider the effects of participating media, such as the scattering of light in the atmosphere.

For modeling the effects of different surface materials on the behavior of light we need introduce the concept of reflectance distribution functions. The reflectance distribution functions capture the appearance of real-world materials. In the most general case, light can enter the surface at one point p_1 , travel in the surface for some distance d and exit the surface at another point p_2 , see Figure 2.4. This behavior is an example of a phenomena called *subsurface scattering*. Subsurface scattering is a mechanism of light transport which describes the behavior of light when it penetrates the surface of a translucent object. The mathematical model for this type of light-surface interaction is called the bidirectional scattering surface reflectance distribution function (BSSRDF) [55, 35]. In order to limit our discussion we do not consider subsurface scattering in more detail in the rest of this thesis. Instead, we focus on a simplified light-surface interaction model, which assumes that light entering the surface at some point p exits at the same point p , thus restricting the set of surface materials \mathcal{M} by excluding translucent materials. The mathematical model for the simplified light-surface interaction model is called the *bidirectional reflectance distribution function* (BRDF) [55]. The BRDF is a fundamental optical property of surface materials. It characterizes the exitant radiance scattered to the hemisphere as a result of incident radiance.

The BRDF $\rho : \mathcal{G} \times \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R}$ is defined for each point x in the set of surfaces \mathcal{G} as the limit of the ratio of exitant radiance L_{out} reflected in the direction $\omega' \in \mathbb{S}^2$ and the incident radiance through the direction $\omega \in \mathbb{S}^2$ as the solid angles containing the corresponding directions tend to zero. In other words, we define

$$(2.12) \quad \rho(x, \omega, \omega') = \frac{L_{\text{out}}(x, \omega')}{L_{\text{in}}(x, \omega) \cos \theta d\omega}.$$

The BRDF function has several important properties, which we will discuss next. Firstly, we note that the BRDF is symmetric with respect to the directions ω and ω'

$$(2.13) \quad \rho(x, \omega, \omega') = \rho(x, \omega', \omega).$$

This symmetry is also known as the *Helmholtz reciprocity*. The BRDF also obeys the law of energy conservation. Intuitively this means that the total reflected power must be less than or equal to the total amount of incident power. The surplus power is considered to transform into some other form of energy, such as heat. The law of energy conservation can also be informally stated as follows

$$\text{light incident at surface} = \text{light reflected} + \text{light absorbed} + \text{light transmitted}.$$

2.4 Mathematical Principles of Realistic Image Synthesis

We have introduced the necessary concepts for quantifying the light transport process and studied how light interacts with surfaces assuming the geometric optics model of light. Next we will present the transport equations governing the propagation of light in a scene. Physically based rendering methods need to have an adequate model for processes that affect light transport, including emission, reflection and finally absorption. The field of radiative transfer has contributed to the theory of realistic image synthesis by providing a set of equations describing these physical processes.

In this section we will introduce the rendering equation and develop some useful mathematical tools for efficient manipulation of the rendering equation.

2.4.1 The Rendering Equation

The *rendering equation* as introduced by Kajiya [36] expresses the light transport in the scene according to the geometric optics model. Solving the rendering equation is equivalent to simulating the physically based behavior of light energy emitted from the light sources in the scene. Although the rendering equation may well not be the final statement on the subject, the importance of a concise mathematical model for realistic image synthesis should not be underestimated. Its generality implies that almost all physically based rendering algorithms can be explained in the mathematical framework which lays itself on the rendering equation [22].

Let X be the space of all real-valued functions defined on $\mathcal{G} \times \mathbb{S}^2$. If $E \in X$ is the *surface emission* function representing the initial lighting configuration then we wish to determine the surface radiance function $L_{\text{out}} \in X$ that satisfies the rendering equation

$$(2.14) \quad L_{\text{out}}(x, \omega) = E(x, \omega) + \int_{\Omega(x)} \rho(x, \omega, \omega') L_{\text{in}}(x, \omega') \cos \theta' d\omega',$$

where E represents the emitted radiance, e.g., E is identically zero for all non-emitting surface points. Assuming that there is no participating media and that light travels in straight lines, one finds immediately that $L_{\text{in}}(x, \omega) = L_{\text{out}}(R(x, \omega), -\omega)$. Intuitively, the meaning of the rendering equation is quite clear; the light energy carried by a stream of photons flowing towards the observer in direction $-\omega$ from the surface point x is simply the sum of emitted radiance and the radiance which is reflected

2.4 Mathematical Principles of Realistic Image Synthesis

from the environment of the point x . The recursive form of the Equation 2.14 expresses that the lighting at the surface point x is dependent of the reflected light from other surfaces. This simply means that the light emitted from the light sources can bounce multiple times in the scene before hitting the observer.

The recursion in the rendering Equation 2.14 can be considered as a limit process in the following algorithmic sense. Assume that we have an observer observing the color L_{out} of the surface point x . The rendering Equation 2.14 tells us that the color could be worked out by stepping in the place of x , looking around the environment $\Omega(x)$ and transforming the observed light in a way characteristic to the surface material ρ at the surface point x . This observing process would continue at the surface points y which are visible in $\Omega(x)$ and so on ad infinitum. In practice, the observing recursion is stopped as soon as the recursive observations do not significantly contribute to the lighting at the starting point x , i.e., the emitted light energy has approximately reached an equilibrium in the scene. It is worth noting that the equilibrium is dynamic in the sense that energy is constantly flowing. The source of the energy could be synthetic (e.g. a light bulb) or natural (e.g. the sun).

James Arvo noted that the mathematical tools provided by *functional analysis* provide a natural way of expressing the equations governing global illumination [5]. Functional analysis is a branch of mathematics concerning the study of abstract vector spaces and the operators that act on those spaces. We will next briefly introduce the relevant concepts of functional analysis and then we will formulate the rendering Equation 2.14 using functional analytical tools. Further details on functional analysis can be found in almost every introductory text on real analysis, see [59] for example.

2.4.2 Vector Spaces

We begin our brief introduction to functional analysis by reviewing elementary properties of vector spaces since these properties will be used throughout this Thesis. We define a real *vector space* as a triple $(X, +, \cdot)$ where X is a set, $+$ is a vector addition operator $X \times X \rightarrow X$ and \cdot is a scalar multiplication operator $\mathbb{R} \times X \rightarrow X$ satisfying the following axioms

- 1. Vector addition is associative:**
For all $x, y, z \in X : x + (y + z) = (x + y) + z$.
- 2. Vector addition is commutative:**
For all $x, y \in X : x + y = y + x$.
- 3. Vector addition has an identity element:**
There exists a *zero vector* $0 \in X$ such that $x + 0 = x$ for all $x \in X$.
- 4. Vector addition has an inverse element:**
For all $x \in X$ there exists an *inverse element* $-x$ such that $x + (-x) = 0$.
- 5. Distributivity holds for scalar multiplication over vector addition:**
For all $\alpha \in \mathbb{R}$ and $x, y \in X : \alpha(x + y) = \alpha x + \alpha y$.
- 6. Distributivity holds for scalar multiplication over field addition:**
For all $\alpha, \beta \in \mathbb{R}$ and $x \in X : (\alpha + \beta)x = \alpha x + \beta x$.
- 7. Scalar multiplication is compatible with multiplication in the field of scalars:**
For all $\alpha, \beta \in \mathbb{R}$ and $x \in X : \alpha(\beta x) = \alpha\beta(x)$.
- 8. Scalar multiplication has an identity element:**
There exists an *identity element* $1 \in \mathbb{R}$ such that $1x = x$ for all $x \in X$.

Given a subset W of X we say that W is closed under vector addition if $\forall x, y \in W : x + y \in W$ and that W is closed under scalar multiplication if $\forall \alpha \in \mathbb{R}, x \in W : \alpha x \in W$. Given a vector space X , a non-empty subset W of X that is closed under addition and multiplication is called a *subspace* of X . The intersection $X \cap Y$ of two vector spaces X and Y is a vector space and the intersection of all subspaces containing a given set of vectors $\{x_1, x_2, \dots\}$ is called its *span*. The span of a set of vectors $\{x_1, x_2, \dots\}$ is denoted by $\text{span}\{x_1, x_2, \dots\}$. We call a set of vectors (x_1, x_2, \dots) *linearly independent* if no vector x_i can be removed from the set without changing its span. Otherwise we call the set *linearly dependent*. A linearly independent set of vectors whose span is X is called a *basis* for X . Although not immediately obvious, we note that every vector space has a basis. We say that the number of elements in a basis of X is the *dimension* of X . The dimension of a vector space can be finite or infinite. A *linear combination* of the vectors (x_1, x_2, \dots) is of the form $\sum_n \alpha_n x_n$, i.e., a weighted sum of the vectors.

Banach space. In order to work with infinite-dimensional vector spaces and to study infinite sums of vectors we are going to introduce a notion of convergence in vector spaces. A real-valued function $\|\cdot\| : X \rightarrow \mathbb{R}$ is called a *norm* given that it satisfies the following conditions:

1. **Positivity.**
 $\|x\| > 0$ for all $x \neq 0$.
2. **Scaling property.**
For all $\alpha \in \mathbb{R}$ and $x \in X : \|\alpha x\| = |\alpha| \|x\|$.
3. **Triangle inequality.**
For all $x, y \in X : \|x + y\| \leq \|x\| + \|y\|$.

We note that the above properties imply that $\|x\| = 0$ if and only if $x = 0$. A vector space equipped with a norm is called a *normed vector space*.

A real valued function $d(x, y) : X \times X \rightarrow \mathbb{R}$ is called a *metric* given that it satisfies the following conditions:

1. **Identity of indiscernibles.**
 $d(x, y) = 0$ if and only if $x = y$.
2. **Symmetry.**
 $d(x, y) = d(y, x)$.
3. **Triangle inequality.**
 $d(x, z) \leq d(x, y) + d(y, z)$.

We note that the above properties imply that $d(x, y) \geq 0$. A norm always induces a metric by $d(x, y) = \|x - y\|$. We call a sequence of vectors (x_1, x_2, \dots) a *Cauchy sequence* if

$$(2.15) \quad \lim_{\min(n, m) \rightarrow \infty} d(x_n, x_m) = 0.$$

Given a normed vector space X , we say that X is *complete* if every Cauchy sequence of X has a limit in X . Completeness of a normed vector space is an intrinsic property of that space. Intuitively this property ensures that there are no "holes" in the space. We call a complete normed vector space a *Banach space*.

2.4 Mathematical Principles of Realistic Image Synthesis

In general there are a multitude of norms to choose from. Two common norms that we will use in this Thesis are the Euclidean norm

$$(2.16) \quad \|x\|_2 := \left(\sum_{i=1}^n x_i^2 \right)^{1/2},$$

and the uniform norm

$$(2.17) \quad \|x\|_\infty := \max\{|x_1|, |x_2|, \dots, |x_n|\}.$$

In the case of function spaces we will make use of *p-norms*, defined by

$$(2.18) \quad \|f\|_p := \left(\int_A |f(x)|^p dx \right)^{1/2},$$

where A is the domain of f . We define the set $\mathcal{L}_p(A)$ as the set of functions defined on the domain A for which the corresponding p -norm exists and is finite. Of particular interest is the set $\mathcal{L}_2(\mathbb{S}^2)$, i.e., the set of square-integrable functions defined on the 2-sphere. This will become more evident as soon as we have equipped the Banach spaces with some geometric structure.

Hilbert space. We will next add more structure to the previously developed real Banach space X . We call a real valued function $\langle x|y \rangle : X \times X \rightarrow \mathbb{R}$ an *inner product* if $\langle \cdot | \cdot \rangle$ satisfies the following conditions:

1. For all $x \in X : \langle x|x \rangle \geq 0$.
2. For all $x, y \in X : \langle x|y \rangle = \langle y|x \rangle$.
3. For all $\alpha, \beta \in \mathbb{R}$ and $x, y, z \in X : \langle \alpha x + \beta y|z \rangle = \alpha \langle x|z \rangle + \beta \langle y|z \rangle$.

An inner product always induces a norm by $\|x\| = \sqrt{\langle x|x \rangle}$. However, the opposite need not be true since there exists norms which can not be defined by means of an inner product. We call a Banach space whose norm is defined by an inner product a *Hilbert space*.

The most important property of Hilbert spaces considering this Thesis is the notion of *orthogonality*. We say that two vectors x and y are orthogonal if they satisfy the following condition:

$$(2.19) \quad \langle x|y \rangle = 0.$$

Orthogonality of vectors x and y is denoted by $x \perp y$. A vector x is said to be orthogonal to a set $W \subset X$ if $x \perp w$ for all $w \in W$. This is denoted by $x \perp W$. One of the most useful applications of orthogonality is *finite basis approximation*. That means that a given Hilbert space X can be approximated by choosing a suitable finite-dimensional subspace W of X and orthogonally *projecting* the

elements of X to W . This finite basis approximation procedure is one of the key ingredients of our novel rendering method.

We call a set of orthogonal vectors *orthonormal* if each vector x in the set is of unit length, that is $\|x\| = 1$. Given an orthogonal set of vectors it is always possible to derive an orthonormal set of vectors spanning the same space. This normalized set of vectors is obtained simply by multiplying each vector by the inverse of its norm, if it exists. Henceforth we shall automatically assume orthonormality if we are given a orthogonal and linearly independent set of vectors.

Given a finite orthonormal basis (e_1, e_2, \dots) for a given Hilbert subspace W of X , the finite approximation procedure reduces to the computation of inner products, i.e., the orthogonal approximation $w \in W$ to $x \in X$ is given by

$$(2.20) \quad w = \sum_n \langle x | e_n \rangle e_n.$$

The orthogonal approximation is also guaranteed to be the closest one with respect to the distance metric induced by the norm of the vector space. The choice of the norm therefore has a great impact on the resulting approximation.

2.4.3 Linear Operators

Now that we have covered the relevant properties of abstract vector spaces we will turn our attention to properties of some transformations acting on these spaces. The most important class of mappings in global illumination is that of *linear* mappings. We call a mapping $\mathbf{T} : X \rightarrow Y$ linear if it satisfies

$$(2.21) \quad \mathbf{T}(\alpha x + \beta y) = \alpha \mathbf{T}x + \beta \mathbf{T}y.$$

for all $x, y \in X$ and $\alpha, \beta \in \mathbb{R}$. In the case of linear mappings we adopt the convention of writing $\mathbf{T}x$ instead of $\mathbf{T}(x)$. This naming convention is based on matrix notation and is justified by the fact that matrices are representations linear mappings. We call linear mappings between function spaces *linear operators*.

Integral Equations. Integral equations can be neatly stated in operator notation. Consider the following integral equation

$$(2.22) \quad f(x) = \int_0^1 k(x, y)h(y)dy$$

Equation 2.22 can be expressed in operator notation as follows

$$(2.23) \quad f = \mathbf{K}h,$$

where \mathbf{K} is the linear operator defined as $\mathbf{K}h(x) := \int_0^1 k(x, y)h(y)dy$.

2.4.4 The Rendering Equation Revisited

In order to fully utilize the previously developed machinery we will next formulate the rendering Equation 2.14 in operator notation. With some exceptions the notation we use is similar to Arvo et. al [6].

We begin by defining the *field radiance operator* \mathbf{G} . The field radiance operator acts on the surface radiance functions by transforming distant surface radiance to local field radiance. As always, we assume that there is no participating media and therefore light travels in straight lines.

The action of this operator is intuitively that of gathering reflected and emitted light from all visible surfaces with respect to a given gathering location. Assuming that L_{out} is a surface radiance function then the field radiance operator can be defined as follows

$$(2.24) \quad \mathbf{G}L_{\text{out}}(x, \omega) := L_{\text{in}}(R(x, \omega), -\omega),$$

where R is the ray-cast operator. It is worth noting that the field radiance operator includes an implicit notion of visibility thereby making the evaluation a computationally costly process.

Next we define the *local reflection operator* \mathbf{R} . The local reflection operator transforms local field radiance into reflected surface radiance according the scattering properties of the surface at the given evaluation point. If L_{in} is a field radiance function, the local reflection operator can be defined as follows

$$(2.25) \quad \mathbf{R}L_{\text{in}}(x, \omega) := \int_{\Omega(x)} \rho(x, \omega, \omega') L_{\text{in}}(x, \omega') \cos \theta' d\omega'.$$

Now that we have defined the field radiance operator \mathbf{G} and local reflectance operator \mathbf{R} we can rewrite the rendering Equation 2.14 in operator notation.

$$(2.26) \quad L = E + \mathbf{RGL}.$$

Equation 2.26 can be written even more compactly by defining a *transport operator* \mathbf{T} as follows

$$(2.27) \quad \mathbf{T} := \mathbf{RG}.$$

This leads to the final version of the rendering Equation 2.26 using operator notation

$$(2.28) \quad L = E + \mathbf{TL}.$$

Neumann Series. We will now introduce a standard tool in functional analysis called the *Neumann series*. The Neumann series allows us to solve the rendering Equation 2.28. Intuitively the Neumann

series can be considered as the operator analogue of the familiar geometric series. By rewriting Equation 2.28 we obtain

$$(2.29) \quad L = E + \mathbf{T}L$$

$$(2.30) \quad \Leftrightarrow (\mathbf{I} - \mathbf{T})L = E$$

$$(2.31) \quad \Leftrightarrow L = (\mathbf{I} - \mathbf{T})^{-1}E = \sum_{n=0}^{\infty} \mathbf{T}^n E = E + \mathbf{T}E + \mathbf{T}^2 E + \dots,$$

where I is the identity operator and \mathbf{T}^n is the notation used for operator composition. The infinite sum is called the Neumann series and the invertibility of the operator $(I - \mathbf{T})$ is a consequence of the laws of thermodynamics and energy conservation. For deeper details about the Neumann series and operator norms see e.g. [59]. We will demonstrate an algorithm for solving global illumination based on the Neumann series in Chapter 4.

Chapter 3

Previous Work

In this chapter we provide an overview of the related previous work. The work in this thesis is mostly based on three different but interrelated categories of previous research, namely those of environment illumination, precomputed transfer and real-time shadowing techniques.

We begin by considering the representation and approximation of environment illumination, i.e., illumination parameterized by direction. While studying environment illumination functions, we make frequent use of the concept of more general spherical functions. By raising the level of abstraction, we are able to analyze the applicability of previous BRDF approximation methods in the context of spherical function approximation. This more general level of study gives insights to the more specific problem of the approximation of environment illumination functions. We also discuss environment map sampling, which can be regarded as a means of approximating the target environment illumination function with directional light sources, and therefore it is relevant in the context of this thesis as well.

In essence, our proposed rendering algorithm for environment illumination is a precomputed transfer method augmented with real-time shadowing techniques. Therefore we review the previous work related to precomputed transfer and real-time shadow algorithms. Combining real-time shadow techniques with precomputed transfer methods has been brought up as an interesting avenue for future work [66].

3.1 Approximation of Environment Illumination

Environment maps [9] achieve a uniform approximation to the environment illumination by storing a sphere of incident radiance at a scene point. In particular, environment maps and other forms of image-based illumination excel at capturing complex real-world lighting environments. This has been one of the key factors for their support in rendering systems. The environment map method [9, 30] is widely used in the context of real-time rendering to account for mirrored reflections due to environment illumination. Environment maps are functions defined on the sphere, so there are a multitude of ways for parameterizing them. A commonly used format is the *cube map*, which is also commonly supported by graphics hardware. A cube map consists of six images, or *texture maps*—where each image corresponds to a face of a unit cube. The faces of the unit cube cover the unit sphere and thus it is possible to store the sphere of incident radiance in a hardware-friendly way. Other parameterizations include the dual paraboloid map [33] and the spherical parametrization

introduced by Blinn and Newell [9].

Environment Mapping Extensions

The basic environment mapping method has been extended with prefiltered environment maps [13, 34, 37, 39] and *high dynamic range* (HDR) imaging [17]. Prefiltered radiance environment map pre-integrates the BRDF with a lighting function to allow for interactive rendering of more complex BRDFs, e.g., BRDFs representing glossy surfaces. High dynamic range imaging is concerned with the acquisition and usage of high dynamic range radiance maps to represent the environment illumination. The motivation for high dynamic range imaging is to accurately represent the continuous intensity distribution found in real scenes. This is accomplished by treating the color channels of an image with floating point values resembling continuum instead of a discrete mapping, say, in the range of 0 – 255. Debevec et al. [16] use high dynamic range radiance maps captured from real-world scenes to illuminate synthetic objects.

BRDF Approximation Methods

Although BRDFs are usually defined as spatially variant 4-dimensional functions ¹, studying the approximation methods for BRDFs gives insight to the 2-dimensional problem of environment illumination approximation as well. Ward [80] presented an approximation model based on a Gaussian stressing its efficiency in data reduction and reproduction. Other researches have looked at deriving *multilobe* representations for the BRDFs. Typically, a multilobe representation is linear combination of radially symmetric spherical functions. The term "multilobe" captures the underlying geometric intuition; some commonly used BRDFs representing surface materials do seem to have multiple "lobes", where the lobes capture the dominant reflectance properties of the surface material at hand. Lafortune et al. [44] used a costly non-linear optimization to fit a set of generalized Phong lobes in order to approximate arbitrary BRDFs. Walter et al. [76] used directional lights and the Phong lighting model to approximate the appearance of a non-diffuse global illumination allowing for interactive rendering with a moving view point. They used a greedy and heuristic algorithm for the fitting of the directional lights. Kautz et al. [37] used a multilobe approximation of the BRDF in the context of prefiltered environment mapping. They compute the approximate representation for the BRDF using a local greedy fitting technique. While not optimal, the greedy technique is more efficient than global optimization and it produces usable results quickly.

Xu and Sun [84] proposed a hybrid representation of a decomposed BRDFs. In their approach the smooth background is represented using a linear combination of spherical harmonics and the sharp lobe using a Gaussian function. Their work differs from ours, however, as Xu and Sun are focusing on the representation of BRDFs. In a similar vein, Hao and Varshney [32] use a hybrid approach for compressing precomputed integrals with spherical harmonics. They compensate for the inadequacy of spherical harmonics for storing "high-frequency" components by storing the high-frequency spikes explicitly. This is justified by the observation that when the approximation is done with truncated spherical harmonic expansions containing only a small number of terms, some of the high-frequency content of the original target function might not be captured by the series expansion. This is best understood by analogy. For example, if we approximate a polynomial of degree n with a polynomial of degree m with $n > m$, some detail is necessarily lost. In comparison to Hao and Varshney, we use a spherical radial basis functions (SRBFs) to represent features of the environment illumination function which are not adequately represented with a finite spherical harmonic expansion containing only a small number of terms.

¹Two dimensions for the incoming light direction and two for the outgoing light direction

3.1 Approximation of Environment Illumination

Environment Map Sampling

Another line of research focuses on the sampling of environment maps [1, 40, 56]. The environment map can be viewed as a collection of directional light sources, by simply considering each texel in the environment map as a directional light source. This is only an approximation, since the texels do span a positive solid angle when projected to the unit sphere, although the approximation error vanishes as the resolution of the cube map increases. In this case, the approximation of the environment illumination is equivalent to choosing a suitable subset of the directional light sources to represent the original environment map. The problem with the point sets generated by stochastic sampling is temporal coherency in sampling dynamic environments, or, rather the lack of it. Temporal incoherency of the sampled point set results in discontinuous shadow boundary jumps, illumination inconsistency and choppy animation unless special care is taken to counter it. Wan et al. [77] propose an adaptive spherical data structure called spherical Q2-tree to alleviate the problem in the context of offline rendering. They take advantage of the local adaptive nature of their data structure which helps to reduce the effects of temporal incoherency. Tamura et al. [73] used a directional light approximation obtained by the method of Kollig et al. [40] of the environment illumination to render photorealistic animations of dynamic scenes in real time using deferred shading to render shadows.

Approximation of Functions on the Sphere

In computer graphics, there exists a multitude of ways to approximate functions on the sphere. The basic idea is to approximate a given target function with some simpler functions suitable for further analysis. The set of functions one should use for constructing the approximation to the target function is usually highly dependent on the application at hand. A basic distinction can be made between linear and nonlinear approximation methods [18]. By linear approximation we mean the process of choosing a suitable finite linear function space defined on the sphere and approximating a given function with the elements of the chosen space. Nonlinear approximation deviates from this approach by not limiting the approximations to come from linear space. A typical problem in nonlinear approximation is to choose a good basis from a given library of bases for approximation. Both of these methods are widely used in computer graphics.

In the context of linear approximation, *spherical harmonics* have received a lot of attention recently. The definition of spherical harmonic functions is quite involved and technical so we will only review the most relevant points regarding this Thesis. Further details about the construction of the spherical harmonic system of functions can be found from most texts on partial differential equations, see e.g. [60]

For our purposes it is sufficient to view the spherical harmonic system of functions as a orthogonal basis for a large class of spherical functions, including the set of all spherical target functions that are relevant in this Thesis. Loosely speaking the spherical harmonic system of functions can be seen as a generalized Fourier system allowing for finite basis approximation. Intuitively the approximation properties of a n-term finite spherical harmonic approximation is similar to that of n-term spherical polynomial, in the sense that they span the same function space.

Dobashi et al. [19] approximate arbitrary luminous intensity distributions by low-dimensional spherical polynomials expressed in the spherical harmonic basis. Ramamoorthi and Hanrahan [58] considered the rendering of diffuse objects under environment illumination. They approximate the environment illumination with a spherical harmonic expansion. They show that only 9 coefficients of the spherical harmonic expansion of the irradiance is enough for low-error results. Their method does not account for visibility, e.g., they are not able to synthesize shadows.

Spherical wavelets were first studied by Dahlke et al. [15] and Freeden and Windheuser [75] and Schröder and Sweldens [62], however Schröder and Sweldens considered primarily the approximation of functions defined on the sphere in the context of computer graphics applications. Schröder and Sweldens construct spherical wavelets on the triangulation induced by a subdivided icosahedron. This construction is independent of the parametrization of the sphere as opposed to the method proposed by Freeden and Windheuser.

Radial basis functions (RBF) are a well known tool in scattered data approximation and interpolation [12]. In this thesis we consider radial basic functions defined in terms of a *kernel* $\phi : \mathbb{S}^2 \rightarrow \mathbb{S}^2$. In addition, we assume that the kernel ϕ is *symmetric, continuous, positive definite* and *zonal*. The kernel ϕ is positive definite if

$$(3.1) \quad \sum_{i,j=1}^N \alpha_i \alpha_j \phi(\omega_i, \omega_j) \geq 0$$

for any $N \in \mathbb{N}$, $\alpha = (\alpha_i)_{i=1,\dots,N} \in \mathbb{R}$ and $\{\omega_1, \dots, \omega_n\} \subset \mathbb{S}^2$, and if the equality in 3.1 holds only for $\alpha = 0$. The kernel ϕ is zonal if for some $\psi : [-1, 1] \rightarrow \mathbb{R}$ we have

$$(3.2) \quad \phi(\omega, \omega') = \psi(\omega \cdot \omega'), \quad \omega, \omega' \in \mathbb{S}^2.$$

If ϕ is kernel satisfying the conditions stated above, then the radial basis function interpolant to a continuous spherical function f usually has the form

$$(3.3) \quad f_N(\omega) = \sum_{j=1}^N \alpha_j \phi(\omega, \Xi_j), \quad \omega \in \mathbb{S}^2,$$

where $\Xi = \{\Xi_1, \dots, \Xi_N\}$ is a set of centers on the sphere \mathbb{S}^2 and the coefficients $\alpha_1, \dots, \alpha_N$ are chosen so that

$$(3.4) \quad f_N(\Xi_i) = f(\Xi_i), \quad \text{for all } i = 1, \dots, N.$$

The interpolant 3.3 provides us an approximation to the original target function f , giving us the freedom to choose the set of centers Ξ . The interpolation condition 3.4 could also be weakened in order to give us more freedom in constructing the approximation.

In our approach we use a combination of linear and nonlinear approximation to better capture the features of the target function while keeping the parameter set characterizing the approximation as small as possible. We use a low order spherical polynomial represented in the spherical harmonics basis together together with a set of radial basis functions to capture the parts of the original function that are not presented well enough by the spherical polynomial. Our approach is greatly inspired by combined polynomial and radial basis function approximations, which have been studied in the field of constructive approximation.

3.2 Precomputed Transfer

Solving the rendering equation in fully dynamic scenes is still far from real-time performance. In order to achieve better image quality in terms of physically based rendering, there exists several techniques that employ some form of precomputation to speed up runtime image synthesis. Pre-computation usually means that some of the degrees of freedom of the original problem are fixed throughout the image synthesis pipeline. In this section we review some precomputation-based techniques in general and precomputed radiance transfer in particular.

3.2 Precomputed Transfer

Image Relighting

Image relighting techniques aim to relight static scenes with dynamic lighting in real-time by fixing the camera position [20]. Image relighting methods use precomputed lighting solutions expressed in some basis to generate relit images as linear combinations of the lighting basis vectors. Malzbender et al. [51] use photographs of a surface under varying lighting conditions to construct a polynomial approximation of the surface lighting and use this approximation to reconstruct the lighting from different directions. Ashikmin and Shirley [7] propose a new set of illumination basis functions, called steerable basis functions for image based relighting. Ng et al. [53] use the Haar system defined over cube map faces together with precomputed lighting solutions expressed in the Haar system to interactively relight the scene at runtime. Although Ng et al. apply compression techniques to reduce their data storage requirements, their method still suffers from large memory costs and long precomputation times.

Light Fields

Light field rendering is a form of image based rendering in which one trades fixed camera and dynamic illumination to dynamic camera and fixed illumination [48, 28]. Light field rendering systems generate different views of a scene from a set of pre-acquired imagery. A surface light field fully describes the radiance in any point and in any direction in the scene. After construction, the surface light field can be used to synthesize different views of the scene in real time. Image based rendering algorithms offer many advantages. The display algorithms for image-based rendering do not require large computing power, real-time rendering is independent of the geometric complexity of the scene and they offer the ability to combine real and synthetic environments together.

Precomputed Radiance Transfer

In this section we give an overview of the work done in precomputed radiance transfer. Generally, PRT methods differ from image-based rendering methods by allowing dynamic lighting and changing the viewpoint, e.g., dynamic camera movement.

As noted by Lehtinen [46] many precomputed radiance transfer methods rely on the following observation. If we assume that the emission function E lies in a low-dimensional function space, then by writing the rendering equation in operator notation we get

$$\begin{aligned} (3.5) \quad L &= (I - \mathbf{T})^{-1} E \\ (3.6) \quad &= (I - \mathbf{T})^{-1} \sum_n \lambda_n e_n \\ (3.7) \quad &= \sum_n \lambda_n (I - \mathbf{T})^{-1} e_n. \end{aligned}$$

By assuming finite-dimensional emission space the above equations say that lighting solution obtained by a given emission vector $(\lambda_1, \lambda_2, \dots)$ is a linear combination of the lighting solutions obtained by using each emission space basis vector e_n as an emission function. This fact allows for the precomputation of these basis lighting solutions $(I - \mathbf{T})^{-1} e_n$ and the dynamic behavior of the emission vector $(\lambda_1, \lambda_2, \dots)$.

Sloan et al. [66] used truncated spherical harmonic expansions to approximate the rendering Equation 2.14. They proposed a real-time rendering method for static diffuse and glossy scenes lit by "low-frequency" environment illumination, i.e., environment illumination represented by a low degree spherical polynomial. As a preprocess, they precompute the transfer functions for each lighting basis function. The transfer functions transform incident lighting into exitant radiance. At runtime, these precomputed transfer functions are applied to actual incident lighting, therefore allowing for dynamic environment illumination. The transfer functions are discretized by point sampling, e.g., point sampling in the triangle vertices. Thus, the transfer function is approximated by hat functions defined over the triangle vertices. Reconstruction can then be efficiently done in hardware by linearly interpolating the vertex values in each triangle. Interestingly, Sloan et al. concluded their work by noting that it would be valuable to combine existing shadowing techniques with their method by decomposing the environment illumination, which is exactly what we are pursuing in this thesis.

The original work of Sloan et al. [66] has spawned a considerable amount of new work on the subject. A lot of research has also been done in order to improve the accuracy of the underlying approximation methods, naturally aiming in increased image quality in the synthesis phase. Spherical function approximation is typically applied to the following items.

- **Environment illumination.** Sloan et al. used low-degree spherical polynomials to represent the environment illumination. The limitations on the accuracy of the approximation, and therefore the lighting environments that can be used, are well known, i.e., practical lighting environments are limited to those that can be represented using a sufficiently low-degree spherical polynomial. One important direction of recent research has been in generalizing the light sources to include local lighting as well as improving the accuracy of the environment illumination approximation.
- **Transfer function.** The transfer function usually incorporates lighting effects such as inter-reflection and self-occlusion. Better approximation can be trivially obtained by increasing the number of transfer sample points. This is by no means the best approach to the problem and it is an active area of research.
- **BRDF.** Due to the limitations of spherical polynomial approximation, a lot of work is done to increase the approximation quality to allow for generalized BRDFs to be incorporated in pre-computed radiance transfer. Using low-degree spherical polynomials, i.e., low-order spherical harmonic basis expansions to approximate lighting transfer including highly glossy materials leads to undesired blurry results. This is a consequence of the used approximation method; we are looking to approximate elements of a space X with elements of a smaller subspace of X , thus the approximation is not necessarily perfect and some features of the original target function are generally lost.

In general, a few trends in the subsequent work can be listed. Namely, removing the restrictions placed on the materials and the lighting. Methods for accomplishing these goals vary widely, from choosing the proper approximation space and method to geometric methods improving the reconstruction of the approximated function.

Sloan et al. [65] improved the PRT method by reducing storage requirements and accelerating its performance by applying compression to the transfer functions. They used clustered principal component analysis (CPCA) to reduce number of transfer function samples required for visually pleasing reconstruction, by partitioning the set of samples into fewer clusters and storing only per-point weights for the original samples and a representative sample for the cluster.

Krivanek et al. [43] also dealt with the problem of discretizing the transfer function, i.e., choosing the set of sample points that faithfully reproduce the original transfer function over the scene. They

3.2 Precomputed Transfer

considered only the hat function basis for triangles and so their method relied on modifying the scene geometry in order to adaptively distribute the set of transfer function sample points.

There are a multitude of extensions to the PRT method trying to improve the support for more complex surface materials. Kautz et al. [38] generalize the PRT method to handle general BRDF shading. They approximate the target BRDF $\rho(x, \omega, \omega')$ by a set of slices $M = \{\rho(x, \omega_i, \omega') : \omega_i \in D \wedge \omega' \in \mathbb{S}^2\}$, where $D \subset \mathbb{S}^2$ is the set of sample directions. The members of the set M are spherical functions which are further approximated with truncated spherical harmonic expansions. This means that they are able to approximate slowly varying directional slices of the original BRDF, corresponding to diffuse and glossy materials for instance. They use dense sampling in the view direction, i.e., the set of sample directions D has 128^2 elements. Curiously, they too conclude that it would be straightforward to extend their approach to decompose the environment illumination into "high and low-frequency terms". As before, a "low-frequency term" is to be understood as a low-degree spherical polynomial.

Lehtinen and Kautz [47] proposed a new method for incorporating anisotropic, glossy BRDFs into precomputed radiance transfer, allowing for real-time rendering performance. They use a new, locally supported basis for calculating the exit radiance for each vertex of the scene. They use the spherical harmonics to represent the environment illumination and the transformation of environment illumination into incident illumination. The incident illumination is then expressed in the new hemispherical basis and multiplied with the BRDF in order to get the exitant radiosity. Using this new basis, together with a simple compression scheme, they are able to demonstrate real-time performance with dynamic lighting and changing viewpoint.

Sloan et al. [67] used a two level approach to precomputed radiance transfer. They simulated the light transport at both a coarse and fine scale and were able to accelerate the run-time performance of the PRT-method. Their method is still limited to smoothly varying environment illumination and subject to limited precision in transfer function approximation, as they assumed the use of low-order spherical polynomials as the approximation device. In a similar vein, Sloan [64] studied the use of normal mapping together with precomputed radiance transfer. Normal mapping can also be considered as a two-level approach to light transport simulation. On a coarse level the incident radiance is obtained from the environment illumination and then a special texture map is used to simulate the lighting effects of small surface details on a fine scale.

Sloan et al. [68] used zonal harmonics (ZH) to model local and quickly-rotatable radiance transfer. Although their method is limited to smoothly varying environment illumination and smoothly varying BRDFs and local light transport effects, they showed that ZHs provide a viable alternative for approximating hemispherical functions. Their method of approximating functions with ZHs is similar that is used in our approach. They fit a multilobe representation to the target function by iterating a single-lobe fitting procedure and extracting the contribution of the optimal lobe from the target function after each step. They optimize in the sense of the L^2 norm, thus possibly averaging important features of the target function. In our approach, we circumvent this problem by using the L^∞ -norm.

Nonlinear optimization approach was taken also by Green et al [29]. They focused on view-dependent sharp highlights caused by non-diffuse BRDFs and detailed, or "high-frequency", environment illumination. They represented the light transport operator using a sum of Gaussian functions. Gaussians on the sphere can practically be considered as locally supported, as they vanish relatively fast according to the distance from the center. Spherical Gaussians are also commonly used in RBF-based approximation and interpolation methods. Green et al. precompute the light transport data for a fixed set of views for all vertices in the scene. Then they apply their fitting method to obtain the sum of Gaussians approximating the precomputed light transport. At runtime, they interpolate the parameters of the closest precomputed views to the dynamic viewpoint in order to obtain the exitant radiance for each vertex in the scene.

Kristensen et al. [42] extend the precomputed radiance transfer to include support for local light sources. Their method relies on a brute-force precomputation followed by compression of the pre-computed data set. They precompute a lighting solution for the scene using a large set of point lights scattered in the scene. They store and compress only the indirect illumination component of the precomputed lighting solutions for each vertex of the scene. At runtime, they use the compressed point cloud to relight the scene lit by dynamic local light sources by reconstructing the precomputed indirect illumination solution. Kontkanen et al. [41] used the Haar-system of orthogonal wavelets to simulate and compress a precomputed indirect illumination solution. At runtime, they project the direct illumination from local light sources to the wavelet basis and reconstruct the indirect illumination solution corresponding to the dynamic lighting configuration. Both Kontkanen et al. and Kristensen et al. thus use a hybrid approach to real-time global illumination for static scenes. They precompute only the indirect illumination component and use standard real-time shading and shadowing methods for the direct lighting. In our approach we use a full precomputed global illumination solution and therefore are able to capture physically accurate smooth shadows caused by direct illumination from distant environment illumination. Admittedly, our method is limited to distant environment illumination.

Sunshine-Hill and Faloutsos [72] also considered local light sources together with a precomputed light transfer solution. Their method is based on a new formulation of the rendering Equation 2.14 simplified for diffuse surfaces. They utilize the observation, that the radiosity transport between two surface points is characterized by a geometric form factor and the surface reflectances. They use the homeomorphism between the set of surface points and a region in \mathbb{R}^2 , e.g., a texture mapping, to discretize the scene in terms of radiosity transport. In effect, they formulate the radiosity problem in the texture space and utilize graphics hardware to speed up the resulting computations.

Ng et al. [53] used the Haar-system over cube map faces to approximate environment illumination and light transport in the context of precomputed transfer. They proposed algorithms for image relighting, i.e., fixed viewpoint, and geometry relighting with changing viewpoint. They precompute a large transfer matrix, by computing a global illumination solution for each directional light represented by a cube map pixel. This procedure results in a huge data set, which they compress by taking advantage of the Haar wavelet transform. They are able to demonstrate real-time performance with both sharp and blurry shadows caused by high-resolution environment illumination map, although their geometry relighting method is limited to diffuse surfaces and their proposed algorithm suffers from large storage requirements and precomputation costs. In their subsequent work, Ng et al. [54] studied triple product integrals. Triple product integrals are commonly encountered in computer graphics, since the rendering Equation 2.14 can be formulated as a triple product integral. Ng et al. [54] were able to overcome the restriction of a fixed viewpoint in geometry relighting, which was one of the greatest limitations in their previous work [53]. While demonstrating images under changing viewpoint and dynamic environment illumination, their method is still far from being real-time, as the renderings took 3-5 seconds per frame. In addition, their precomputation times and storage requirements remain high.

The work of Ng et al. [53, 54] was followed by Wang et al. [78] and Liu et al. [49]. They both used a separable BRDF approximation in combination with the nonlinear wavelet approximation in light transport to allow for interactive frame rates in rendering non-diffuse objects under detailed environment illumination. In a later work, Wang et al. [79] proposed an extension to their previous work to include support for the full BSSRDF model.

Recently, Tsai and Shih [74] formulated precomputed radiance transfer in terms of spherical radial basis functions (SRBF), also known as zonal basis functions (ZBF). Tsai and Shih used uniformly distributed ZBFs for approximating the transfer functions and nonlinear, scattered basis for approximating the lighting environment, thus requiring a change of basis before the functions are in the same space for efficiently computing their inner product for evaluating the exitant radiance.

3.3 Real-Time Shadow Techniques

Shadows provide important visual cues about the spatial relationships of the objects in the scene. However, rendering robust shadows in the context of real-time rendering is not an easy task. In this section we review two of the most commonly used methods for real-time shadow generation for point light sources [14, 83, 4].

Shadow Maps

Shadow mapping [81] is a conceptually simple algorithm for real-time shadow generation. The basic algorithm consists of two passes. In the first pass, a depth map is computed from the light's point of view. This depth map is also called the shadow map. In the second pass the scene is rendered from the camera's point of view and a visibility test is applied to each rasterized fragment. For each fragment, the visibility test compares the fragment's distance to the light source and the value in the shadow map. The fragment is visible if it is closer to the light source, i.e., the fragment's distance to the light source is smaller than the value in the shadow map. Shadow mapping has numerous advantages. It can be applied in general situations, i.e., it has no limitations regarding the geometry of the objects in the scene. Shadow mapping is relatively easy to implement and due to hardware support, it is fast as well. Aliasing artifacts caused by undersampling the depth field from the light source are a major nuisance in shadow map based shadow algorithms. A lot of research is devoted to overcome the aliasing artifacts [10, 26, 63, 69, 85, 82, 52, 2]. Another source of problems is the numerical imprecision of the shadow maps. This results in false visibility results in places where the rasterized fragment's distance and the shadow map value are approximately the same.

Shadow Volumes

The original shadow volume algorithm by Crow [14] is a geometry based algorithm for shadow determination. It has recently received a lot of attention [23, 11, 50, 3] due to the advances in rendering hardware. The algorithm operates on the geometry of the objects in the scene. In the first pass, all the silhouette edges of the geometry with respect to the light source are found. Then the silhouette edges are extended in the direction away from the light source to form a set of quadrilaterals. The shadow determination is motivated by a simple geometric consideration. A scene point is in the shadow only if it lies inside the bounded volume spanned by the set of quadrilaterals. Hence the term shadow volume. The shadow volume algorithm is amenable for hardware implementation by taking advantage of hardware stencil buffers. The drawbacks of the shadow volume algorithm are costly silhouette edge determination and shadow volume rendering, which uses up a lot of the available fill rate especially in highly occluded scenes.

Chapter 4

A Hybrid Algorithm for Environment Illumination Rendering

In this chapter we present a hybrid algorithm for environment illumination rendering. The proposed algorithm combines the strengths of previously introduced rendering methods, namely those of precomputed radiance transfer and shadow mapping. Precomputed radiance transfer excels in reproducing slowly varying lighting effects induced by distant environment illumination. On the other hand, PRT methods require a prohibitive amount of computational resources in order to reproduce lighting effects from high-intensity light sources with small size. Guided by this simple observation, we propose to use shadow mapping for these cases. Thus the problem of rendering environment illumination lit scenes has been transformed to the problem of decomposing the environment illumination into suitable parts to be handled with previous techniques.

The proposed rendering algorithm is applicable for real-time rendering of static scenes lit by distant environment illumination. In contrast to previous methods, our method requires a modest amount of resources while providing visually pleasing results. In essence, we trade physical accuracy for computational resources. That is, we use approximative methods suitable for GPU implementation for computing the direct illumination from point-like light sources. We begin with an overview of the phases of the algorithm after which we study each of the phases in greater detail. Results are discussed later in Chapter 5.

4.1 Overview of the Algorithm

Our proposed rendering algorithm consists of three different phases, precomputation, decomposition and rendering. Precomputation is done as an offline process similarly to previous precomputed radiance transfer methods. We precompute the transfer functions at a set of sample points and subsequently approximate them with low-degree spherical polynomials. Decomposition and rendering is done at runtime to allow for dynamic environment illumination. The decomposition of the environment illumination is done once per given environment illumination function. In the case of dynamic environment illumination, the decomposition is done for each frame. In the decomposition step we analyze a given environment illumination function E and decompose it to the form

$$(4.1) \quad E(\omega) \approx \mathbf{P}E(\omega) + \sum_j \lambda_j \phi(\omega, \Xi_j, R_j),$$

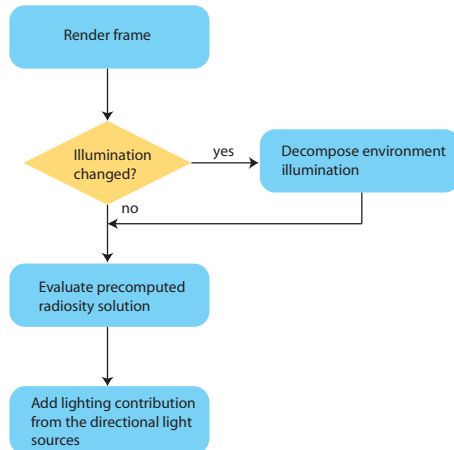


Figure 4.1 A flowchart of per-frame computations.

where \mathbf{P} is the projection operator to a low dimensional spherical function space, ϕ is a spherical radial basis function with radius R_j , center Ξ_j and weight λ_j for each j . The decomposition is an approximation of E . After we have obtained the decomposition we render the scene. Since light transfer is a linear process, we can use an additive multipass technique for the rendering.

Firstly, the projection operator \mathbf{P} is used to evaluate the precomputed lighting basis solutions, so that \mathbf{PE} can be regarded as the slowly varying part of the environment illumination. After evaluating the precomputed transfer functions, a first pass of the scene is rendered. In effect, the first pass synthesizes global illumination effects due to slowly varying environment illumination.

Secondly, a shadow map is created for every radial basis function $\lambda_j\phi(\omega, \Xi_j, R_j)$ approximating a sharp lobe in the original environment illumination. Finally, we render the scene with the shadow maps, using additive blending to accumulate the lighting as in the Equation 4.1. A flowchart of the per-frame computations is shown in Figure 4.1. We continue our discussion by considering each of the three phases in more detail.

4.2 Precomputation Phase

In the precomputation phase we compute the lighting solutions corresponding to the choice of the basis for the emission space at a given set of sample points. Then apply a linear finite basis approximation method to obtain an efficient representation for the lighting solutions. Since the precomputation step essentially requires solving the rendering Equation 2.14 in a set of sample points, it is computationally the most expensive phase of the algorithm.

We approximate the lighting solutions with a low-dimensional function space. As previously noted in Chapter 3, we use the spherical harmonic orthogonal system of functions. Furthermore, we saw in Chapter 2 that the best approximation by an orthogonal system of functions is obtained via orthogonal projection. This approximation is best in the sense that it minimizes the distance metric of the target function and the approximant induced by the norm of the underlying Hilbert space.

The orthogonal projection \mathbf{P} itself is characterized by an orthonormal basis of low-dimensional sub-

4.3 Decomposition of the Environment Illumination

space M , that is, if $M = \text{span}\{e_1, e_2, \dots, e_n\}$, then

$$(4.2) \quad \mathbf{P}x = \sum_{k=1}^n \langle x | e_k \rangle e_k.$$

Since the spherical harmonics, denoted by $Y_{l,m}$ form an orthonormal basis for a large class of spherical functions, the orthogonal projection \mathbf{P} is known. Therefore, in order to construct the approximation to the transfer functions we need to be able to evaluate the inner products $\langle x | Y_{l,m} \rangle$, which in this case are product integrals over the sphere \mathbb{S}^2 .

There are many possible approaches to the numerical computation of the required spherical integrals. We applied a standard Monte-Carlo method to the integration due to its simplicity. We compute a full radiosity solution for each lighting basis function by utilizing the method of truncated Neumann series, see Chapter 2 for more details on the Neumann series.

Pseudocode for the precomputation algorithm is given in Figure 4.2. Procedure `PRECOMPUTE-RADIOSITY` computes the final radiosity solution by applying the Neumann series method. First, we compute the direct lighting due to environment illumination (lines 5–27). Second, we use the direct lighting to propagate and accumulate the indirect lighting resulting from subsequent bounces of the direct lighting (lines 28 – 50). Procedure `MAKE-HEMISPHERICAL-HAMMERSLEY-SAMPLES` generates a set of pseudorandom samples on the hemisphere $\Omega((0, 0, 1))$. These samples are used to estimate the expected value of the radiosity integral. The direct lighting is computed by fixing the environment illumination $E(\omega) = \sum_{l,m} Y_{l,m}$, i.e., lighting the scene with spherical harmonic basis functions and applying standard Monte-Carlo integration method. The indirect illumination is computed by iteratively interpolating the multiply reflected light bounces and accumulating the result. Procedure `RAY-CAST(p, d)` casts a ray $p + \lambda d, \lambda > 0$ and returns the interpolated radiosity at the hit point or zero depending whether the ray hit the scene or not.

4.3 Decomposition of the Environment Illumination

The runtime part of the algorithm consists of the decomposition of the environment illumination followed by the rendering of the scene. We begin our discussion by considering the motivation behind the decomposition, after which we analyze the desired properties of the decomposition as a tool for increasing the accuracy of previous precomputed radiance transfer method. We continue by describing the proposed decomposition model and then we present the decomposition algorithm in more detail. We conclude this section by considering some issues related to the implementation.

The idea of decomposing the environment illumination into components describing with different characteristic features has been brought up before as a promising direction for future research [66, 38, 53]. In spite of this attention, we are not aware of any previous work which has successfully attacked this problem. Real-world illumination provides countless examples of features which can be intuitively classified by their distinctive features. For example, a typical natural outdoor illumination environment is lit by the sun and the sky. The sun is a small concentrated light source, which gives rise to the hard shadows often found in clear-sky outdoor imagery. On the other hand, the sky itself, which evidently varies slowly on the hemisphere, contributes to the general hue of the scene. Midday sky is sincerely blue, whereas at sunset the sky exhibits a wider spectrum of colors, ranging from reddish orange all the way to dark blue. Indoor scenes are no exception in this case. For example, a typical indoor scene contains an architectural model of a room or a hall lit by certain artificial light sources. Indirect, or reflected illumination from the light sources is smooth. The smoothness of the indirect illumination is evident, considering the averaging process that governs its behavior. On the other hand, some of the artificial light sources may have small surface area, e.g., light bulbs, and

```

procedure PRECOMPUTE-RADIOSITY(int num_samples, int bounces)
1 // Definitions
2 float C ← 2ρ/num_samples // Normalization factor
3 float Ylm[] // Basis function evaluation
4 Vec3[] samples ← MAKE-HEMISPHERICAL-HAMMERSLEY-SAMPLES(num_samples)
5 for each vertex i of the mesh do // Compute direct lighting
6   Vec3 normal ← GET-VERTEX-NORMAL(i)
7   Vec3 pos ← GET-VERTEX-POSITION(i)
8   Mat3 transform ← BUILD-ORTHONORMAL-BASIS(normal)
9   for each sample s in samples do
10    Vec3 sample_dir ← MATRIX-MULTIPLY(transform, s)
11    float V ← VISIBILITY(pos, sample_dir) // Evaluate visibility
12    float H ← MAX(0, normal · sample_dir) // Evaluate BRDF
13    EVALUATE-SPHERICAL-HARMONIC-ARRAY(sample_dir, Ylm)
14    for each basis function j do // Update estimates
15      float update ← V * H * Ylm[j]
16      Vec3 rgb(update, update, update)
17      incident_radiance[i].basis[j] ← incident_radiance[i].basis[j] + rgb
18    end for
19  end for
20 end for
21 for each vertex i of the mesh do
22   for each basis function j do
23     exitant_radiance[i].basis[j] ← C * incident_radiance[i].basis[j]
24     accumulated_radiance[i].basis[j] ← exitant_radiance[i].basis[j]
25     incident_radiance[i].basis[j] ← 0
26   end for
27 end for
28 for each bounce b in bounces do // Compute indirect lighting
29   for each vertex i of the mesh do
30     Vec3 normal ← GET-VERTEX-NORMAL(i)
31     Vec3 pos ← GET-VERTEX-POSITION(i)
32     Mat3 transform ← BUILD-ORTHONORMAL-BASIS(normal)
33     for each sample s in samples do
34       Vec3 sample_dir ← MATRIX-MULTIPLY(transform, s)
35       Vec3[] I ← RAY-CAST(pos, sample_dir) // Interpolate radiosity
36       float H ← MAX(0, normal · sample_dir) // Evaluate BRDF
37       for each basis function j do
38         incident_radiance[i].basis[j] ← incident_radiance[i].basis[j] + H * I[j]
39       end for
40     end for
41   end for
42 for each vertex i of the mesh do // Normalize and accumulate radiosity
43   for each basis function j do
44     float B ← C * incident_radiance[i].basis[j]
45     exitant_radiance[i].basis[j] ← B
46     accumulated_radiance[i].basis[j] ← accumulated_radiance[i].basis[j] + B
47     incident_radiance[i].basis[j] ← 0
48   end for
49 end for
50 end for

```

4.3 Decomposition of the Environment Illumination

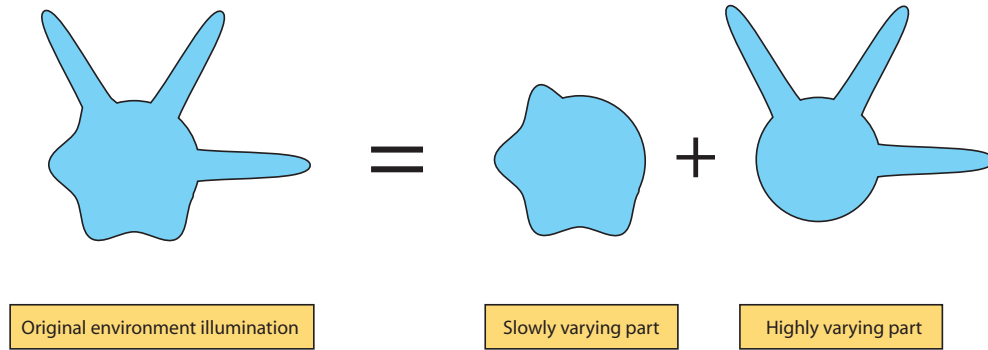


Figure 4.3 The concept of two-scale decomposition for spherical functions.

they may thus be considered as idealized point light sources without introducing visually significant error to the resulting image. The previous examples make it intuitively clear, that the decomposition problem is indeed a meaningful one. In addition, studies of real-world illumination [21] have confirmed this intuitively clear phenomenon.

4.3.1 Decomposition Model

We propose a two-scale decomposition of the environment illumination. The distinctive nature of the desired features in the two-scale decomposition follow from the approximation properties of the function spaces which are chosen for each scale. In other words, given two approximation spaces M_1 and M_2 , we are looking an element x in the space $M_1 \oplus M_2$. The concept of two-scale decomposition is shown in Figure 4.3. In our approach we use the space of low-degree spherical polynomials spanned by the spherical harmonics as the approximation space for the slowly varying component of the environment illumination, since they fit in rather nicely to the precomputed radiance transfer framework. The choice of approximation space for the highly localized features is naturally guided by our goal of using commodity graphics hardware to accelerate the lighting calculations. Therefore it would be advantageous to use such an approximation space, with elements that can be satisfactory approximated with the standard directional light model augmented with shadow maps. Guided by these observations, we are now able to summarize three of the most desirable features of the sought approximation space.

- **Localization of support.** In the two-scale decomposition we are looking for the concentrated light sources with small solid angle, e.g., spherical measure. These features are local by definition, and so they should be best captured with functions with local support. This is in contrast to global, or slowly varying features.
- **Positivity.** Positivity fits nicely to the directional light model, which we use, in the sense that the results of the decomposition are easily interpreted when rendering the scene.
- **Computational efficiency.** Since we are constructing a real-time rendering algorithm, some consideration has to be done regarding the efficiency of the chosen approximation space. Ideally, we would like to have functions, that are efficient to evaluate and to store.

It turns out that spherical radial basis functions fill these requirements rather nicely. Therefore we adopt the use of SRBFs for the approximation of highly intensive localized light sources. To summarize, our two-scale decomposition of a given environment illumination function E consists essentially of a low-degree spherical polynomial plus a finite sum of SRBFs, as shown in Equation 4.1. We denote the space spanned by a low-degree spherical polynomials M_1 and the space spanned by SRBFs with a fixed radial basis function M_2 . The chosen decomposition model fits nicely by design to the rest of the proposed rendering framework.

4.3.2 Strategies for Constructing the Approximation

Now that we have chosen the approximation spaces M_1 and M_2 there remains the question of choosing best approximation \tilde{E} to E from the space $M_1 \oplus M_2$. The quality of the resulting approximation should be evaluated, not solely in terms of the distance $\|E - \tilde{E}\|$ in some norm, but with respect to the quality of the image obtained using the approximation \tilde{E} . We continue our discussion by considering possible approaches for constructing the element \tilde{E} .

SRBF Expansion

The space M_2 spanned by spherical radial basis functions is infinite dimensional and therefore we should construct a finite, possibly nonlinear subspace of M_2 in order to apply numerical computing methods. One way of constructing a nonlinear subspace is to select a finite set of centers for the radial basis functions. In our method the set of centers is chosen greedily from the set of local maximums. This strategy follows the intuition that concentrated light sources are actually local maximums of the environment lighting function.

From here on, we are assuming that we have chosen M_1 and a finite set of radial basis functions spanning the space \tilde{M}_2 .

Minimization of the \mathcal{L}_2 -norm

Suppose that we are trying to find the best approximation to E by minimizing the distance $\|E - \tilde{E}\|_2$, where $\tilde{E} \in M_1 \oplus \tilde{M}_2$. Now, the problem is well posed and can be solved easily. We simply need to discretize the resulting integral equations by approximating the integral by a finite sum and solve the resulting set of linear equations. This least squares procedure is a well known optimization method. One possible source for problems is the fact that the parameter space for the radial basis functions is unrestricted. This can, and in our experience will lead to negative coefficients in the lighting via negative SRBF weights. To satisfy the positivity constraint, the parameter space can be restricted to a convex subset of the \mathbb{R}^n , effectively enforcing positive SRBF weights. The resulting nonnegative least squares problem is also well studied and can be solved by standard methods. We experimented with the nonnegative least-squares method only to find out that it does not give satisfactory solutions to the decomposition. The rationale behind our experimental results is, that the polynomial part P captured almost all the energy in the target function, i.e., the directional lights derived from the SRBFs carried insignificant contribution to the resulting image. The problem with \mathcal{L}_2 norm is that the optimization methods result in an averaging process which does not enforce our intuitive notion of highly localized light sources. In other words, minimizing the average distance $\|E - \tilde{E}\|_2$ is not enough, since there may still be highly local peaks in the residual. Therefore, using the \mathcal{L}_2 norm is not the right approach for approximating features which depend on fast pointwise convergence of the approximating function, and this is essentially the case we are dealing with.

4.3 Decomposition of the Environment Illumination

A Combined Method for Approximation

We have shown, that using approximation methods based on the $mathcal{L}_2$ norm minimization does not yield satisfactory results, since it is apt to miss precisely those high-intensity localized spikes in the original environment illumination that we would like to approximate with directional light sources. Therefore we propose a combined approach for constructing the approximant. Suppose that E is the original environment illumination function and $P \in M_1$ is a low-degree spherical polynomial obtained by orthogonally projecting E into M_1 . Let's consider the pointwise distance $|E(u) - P(u)|$, for a moment. It is intuitively obvious, that the quantity $|E(u) - P(u)|$ obtains large values in high-intensity points u that are not well approximated by a low-degree spherical polynomial. This has direct influence on the final image quality. The precomputed transfer solution is parametrized by the environment illumination represented by P and therefore it cannot exceed its accuracy in reproducing the illumination effects of the original lighting environment E . On the other hand, we should be able to increase the resulting image quality by approximating a small neighborhood $B(u, R)$ of u with a constant function with value $E(u) - P(u)$ and considering it as a directional light source. Guided by these observations we propose an approximation strategy, which is based on minimizing the sup-norm $\|E - \tilde{E}\|_\infty = \sup_{u \in \mathbb{S}^2} |E(u) - \tilde{E}(u)|$. This minimization process leads to a greedy algorithm for the construction of the approximation \tilde{E} .

Before we present our algorithm in detail we will consider some of the assumptions and claims we made regarding the effectiveness of the combined approximation strategy in more detail. Firstly let's consider the increase in image quality. We suggested that by approximating the neighborhoods of the largest pointwise deviations with a set of directional light sources, we should expect to achieve an increase in image quality. However, while we are able to approximate the original environment illumination more accurately, we are still using an idealized light model in rendering. This means that we are essentially collapsing the neighborhoods to a single point in rendering. We acknowledge that this is a potential source for problems in the sense that we are leaving the theoretical safe haven of physically based rendering and stepping into the dangerous realm of real-time tricks and techniques. This, however, is the case with almost all rendering methods suitable for real-time applications, and therefore we do not feel that this fact substantially hinders the usability of the proposed rendering algorithm. Secondly, we suggested that all the neighborhoods with large pointwise deviations are to be approximated with directional light sources. That is not the whole truth however, for we assumed that the polynomial part P is in fact the slowly varying part of the original function. This is not the case, if we simply apply an orthogonal projection of E into M_1 since the original environment illumination may still contain high-intensity spikes which may give rise to the so called *Gibbs* effect in the resulting polynomial P [27]. This is due to the fact, that we are seeking the closest polynomial to match the original function, regardless to the fact that we are not trying to match the original environment illumination function *per se*, since we are trying to match the slowly varying component of it. Therefore, we propose to use a filtered version of the original environment illumination function as a target for approximation by low-degree spherical polynomials, in order to reduce the distance between the target function E and the approximation space M_1 .

4.3.3 Decomposition Algorithm

Now that we have developed the decomposition model and chosen a suitable strategy for the actual decomposition process, we are quite ready to present the decomposition algorithm in a more detailed fashion. The algorithm accepts a cube map representation of the original environment illumination as an input and outputs a finite spherical harmonic expansion of the spherical polynomial corresponding to the slowly varying component of the original illumination function and a set of directional light sources approximating the localized, high-intensity features, if present in the original

function, which are not adequately captured by the polynomial component of the decomposition.

The cube map contains radiance values for three wavelengths, namely those of red, green and blue. We call these radiance triples simply *colors*. We need to analyze the luminous intensity of spherical illumination functions and therefore we use a luminous intensity function I to convert the radiant energy to luminous, i.e., visible energy corresponding to the sensitivity of the human eye. To each color c we can assign a luminous intensity value by defining a luminous intensity function $I : \mathbb{R}^3 \rightarrow \mathbb{R}$, which maps $I(c) = \lambda \cdot c$, where λ is a suitable weight vector. From now on, we assume that this luminous intensity conversion has implicitly been done in cases where we refer to scalar valued spherical functions.

The decomposition algorithm consists of the following four steps.

1. Sample the original environment illumination.
2. Construct the slowly varying component of the sampled environment illumination.
3. Compute the SRBF expansion representing the localized, high-intensity features of the original environment illumination.
4. Construct the directional light approximation from the SRBF expansion for rendering.

We will now discuss each of the steps in more detail.

Spherical Function Sampling

The purpose of this step is to construct a smooth spherical function representing the original environment illumination given as a cube map. The cube map representation was chosen since it is simple and widely supported by customer-level rendering hardware.

The original environment illumination cube map itself can be obtained by several different ways. For instance, we can use high dynamic range illumination maps obtained from real-world locations. To account for dynamic illumination we can use precaptured video sequences or alternatively we can sample the dynamic environment illumination directly from the scene by rendering a cube map at the desired environment illumination sample point. Proper sampling of the scene is crucial in order to avoid aliasing effects in the dynamic cube map. In order to keep the consumption of runtime resources within reason, we have to limit the sampling density, i.e., the cube map resolution. This might cause aliasing artifacts due to undersampling. To overcome the aliasing artifacts we employ a supersampling procedure to the environment map. Supersampling is implemented by rendering the faces of the cube map to a larger resolution than the target and applying an averaging box filter to the supersampled cube map to downsample the cube map to the desired resolution.

After we have obtained the environment illumination cube map we smooth it by convolving the cube map with a spherical gaussian kernel. This results in a smooth environment illumination function represented by a cube map. This smoothing operation has two purposes. Firstly, the original environment illumination data might contain outliers due to artifacts related to the data measurement process. Secondly, the smoothed data set is better suited for analysis, since we have all the tools developed in differential calculus at our disposal.

We use an efficient algorithm for computing the convolution of the cube map and a spherical kernel with small support. The efficiency is achieved by culling the cube map texels which fall outside the bounding box of the kernel support. The bounding box is computed by projecting the

4.3 Decomposition of the Environment Illumination

cone of directions containing the support of the smoothing kernel to the faces of the unit cube. This projection is fast and it allows us to discard a large amount of texels which reside outside the bounding box. After eliminating a lot of useless texels, we process each texel t in the bounding box separately. If the texel t is contained in the support of the kernel we increment our integral accumulator, otherwise we discard the texel. As an end result we have obtained an efficient method for computing spherical convolution over cube map texels.

The cube map representation has some disadvantages however. The solid angles spanned by the cube map texels are not uniform over the sphere. This is easily verified by comparing the solid angle spanned by texels in the corners of the cube versus the texels at the centroids of the cube faces. In addition, since the cube map consists of six separate textures, continuity across the edges is a problem. To overcome the difficulties arising from the cube map representation we use a spherical data structure derived from a regular icosahedron. The data structure is constructed by subdividing the equilateral triangles of the regular icosahedron into four children. This process can be repeated for each child until a desired resolution is obtained. The resolution can be controlled for example by requiring certain covering properties from the induced point set, e.g., the subdivision process is iterated until the maximum of the minimum distances between the triangle vertices is below some threshold. The regular icosahedron suits our purposes well, since its faces are equilateral triangles with uniform area. In addition, the discrete topology is well established in the sense that neighborhood algorithms are simple. The regular icosahedron is well suited for adaptive algorithms as well, where the adaptation is done by refining the local subdivision level at a point of interest. The resulting data structure can be seen as spherical quadtree with uniform subdivision [24, 25, 45].

We sample the cube map at the vertices of the subdivided icosahedron. We assume that the icosahedron is subdivided so that the sampling is dense enough not to introduce further aliasing artifacts to the approximation.

Computing the Spherical Harmonic Expansion

We now have reconstructed the original environment illumination function by point sampling at the vertices of the subdivided icosahedron. This reconstruction preserves all the important features of the original function and discards outliers since we perform a smoothing step.

To obtain the slowly varying part of the reconstructed environment illumination function, we project the function to the space spanned by spherical harmonics of a finite degree. The projection is done by computing the inner products $\langle E | e_j \rangle$ for each basis function e_j resulting in a n -dimensional coefficient vector. The inner products are simply product integrals over the sphere and we calculate the inner products using a simple quadrature rule defined over the cube map texels. The integration is symmetric with respect to the faces of the cube map. Therefore the algorithm for computing the inner products is same for each face of the cube map. Pseudocode for the computation of inner products is shown in Figure 4.4.

In order to damp the ringing effects caused by this finite series expansion, we apply an additional smoothing step so as to decrease the distance between the environment illumination function and the approximation space. The smoothing kernel is defined using the spherical harmonic basis, therefore allowing for fast convolution using inner product properties. In this case, the smoothing corresponds to a dot product in n -dimensional Cartesian space \mathbb{R}^n . This procedure is sometimes called windowing and it is a commonly used technique in signal processing. We argue that the smoothed version of the sampled environment illumination is better captured by the spherical harmonic expansion, since the windowing process is designed to eliminate those high-frequencies, or concentrated light sources if you wish, of the original signal which are not well enough captured by the low-degree spherical polynomial represented in the spherical harmonics basis.

```

procedure COMPUTE-INNER-PRODUCT (Vec3[] * sh_coef, Vec3 P, Vec3 D)
1  float [] Ylm
2  for each y in 0..cubemap.size - 1 do
3    for each x in 0..cubemap.size - 1 do
4      Vec3 pos ← P + (y + 0.5) * dy + (x + 0.5) * dx
5      float r ← LENGTH(pos)
6      NORMALIZE(pos)
7      float w ← da * DOT(pos, D) * 1/r2
8      EVALUATE-SPHERICAL-HARMONIC-ARRAY(pos, Ylm)
9      Vec3 rgb ← GET-COLOR(cubemap.face, x, y)
10     for each basis function i do
11       sh_coef[i].x ← sh_coef[i].x + w * rgb.x * Ylm[i]
12       sh_coef[i].y ← sh_coef[i].y + w * rgb.y * Ylm[i]
13       sh_coef[i].z ← sh_coef[i].z + w * rgb.z * Ylm[i]
14     end for
15   end for
16 end for

```

Figure 4.4 Pseudocode for computing the inner products of cube map faces and spherical harmonic basis functions using a simple numerical quadrature rule. For each cube map face there is a corresponding direction D and a origin P . The vectors dx and dy correspond to raster directions and they are defined on the plane of the corresponding cube map face. The scalar da is the area of one cube map texel. The procedure LENGTH(Vec3 x) simply returns the length of the vector x . The procedure NORMALIZE(Vec3 x) returns x divided by its length. The procedure EVALUATE-SPHERICAL-HARMONIC-EXPANSION(Vec3 d , float [] Ylm) evaluates the spherical harmonic basis functions in the direction d and stores the result of the evaluations in the array Ylm . The procedure GET-COLOR(CubemapFace $face$, int x , int y) returns the rgb-color of the texel with coordinates x and y .

After we have the smoothed spherical harmonic expansion of the original sampled environment illumination we reconstruct the spherical function on a subdivided icosahedron similarly as in the previous sampling step. This icosahedral reconstruction step is done to allow for fast pointwise difference calculations in the following steps.

Computing the SRBF Expansion

The next step in obtaining the decomposition is to construct the SRBF expansion representing set of concentrated, high-intensity light sources in the original environment illumination. Given a radial basis function ϕ , the SRBF expansion is uniquely determined by a set of pairs of the form $(\Xi_i, R_i), i = 1, \dots, k$, where $\Xi_i \in \mathbb{S}^2$ represents the center and R_i represents the radius of the given radial basis function ϕ . We observe that each concentrated light source is a local maximum of the environment illumination function. This simple observation is used to construct the initial SRBF expansion centers. This initial set consists of possible candidates for point light approximation. After we have chosen the set of SRBF centers we are to determine their corresponding radiuses. This can be done with a multitude of ways. First, we could deduce a common radius for the SRBFs from the point light approximation model. That is, we can use the point light approximation model to approximate light sources that cover some small positive solid angle on the sphere instead of the

4.3 Decomposition of the Environment Illumination

zero solid angle of spherical points. Second, we could infer the proper radius for each SRBF (Ξ_i, R_i) from the neighborhood of the local maximum corresponding to the center Ξ_i . This neighborhood can be thought of as the end result of a hill-climbing process on the subdivided icosahedron. The hill-climbing process works as follows. For each vertex v on the icosahedron, we follow the path from v towards neighbors with increasing values until we hit a local maximum Ξ_i . We call the set of subdivided icosahedron vertices that converge to Ξ_i , a *local maximum neighborhood* of the center Ξ_i . The problem with local maximum neighborhoods is that they are not symmetric with respect to their corresponding centers. Since SRBFs are symmetric with respect to their centers, there is generally no trivial way to deduce a proper radius from the local maximum neighborhoods. Third, we can apply a nonlinear optimization method for optimizing the radius in the sense of some norm $\|\cdot\|$. A naive optimization method would first discretize the parameter space by choosing a finite set of parameter values and determining the optimal parameter from the set. Fourth, we could make use of the differential geometric properties of the surface induced by the spherical illumination function. We could relate the radius of the SRBF with center Ξ_i with the curvature of that point, since the curvature is related to the radius of the sphere that locally matches the curvature at point Ξ_i . In practice however, we have found that although all the methods described above give useful results, the simplest one works quite well. Thus, we use a fixed radius for each SRBF center Ξ and therefore the problem at hand is reduced to the problem of finding suitable centers Ξ .

By choosing the centers Ξ_i of the SRBF expansion to coincide with the vertices of the subdivided icosahedron we are in fact introducing some approximation error, since the subdivided icosahedron consists of a finite number of points. This naive assignment might cause temporal aliasing artifacts, due to the discrete nature of subdivided icosahedron vertices. For instance, consider an environment illumination consisting of a single small-area light source that is rotating around the scene with respect to time. If we assume that the solid angle spanned by the light source is smaller than the solid angle covered by any one-ring neighborhood of a vertex on the uniformly subdivided icosahedron, it becomes obvious that the local maximum vertex on the icosahedron changes by discrete jumps while the path of the light source is continuous.

We observe that the true, or the continuous maximum lies in the one-ring neighborhood of the corresponding discrete maximum vertex. To address the continuity required to counter temporal aliasing artifact we introduce a sort of super-sampling process to accurately evaluate the location of the local maximums. More precisely, we approximate the surface in a small neighborhood of a local maximum vertex with a quadric. The quadric fitting is done by projecting the icosahedron vertices near the local maximum to a plane with normal vector pointing to the local maximum direction and then applying a standard linear least-squares method to find out the quadric surface coefficients. The quadric surface has a simple analytical form which can be used to derive a more accurate local maximum for the original surface. The resulting center points Ξ_i are then adjusted to coincide with these more accurate maximums.

Constructing the Directional Light Approximation

We have now obtained the SRBF expansion representing the concentrated, high-intensity light sources of the original environment illumination. We have also obtained a low-degree spherical polynomial representing the slowly varying, or low-frequency part of the original environment illumination. The next step is to construct the directional light approximation from this decomposition.

The directional light model is derived from the properties of current rendering hardware, such that we are able to use hardware acceleration to enhance the performance. More precisely a directional light is a pair (d, r) , $d, r \in \mathbb{R}^3$, where d is incoming light direction and r is the radiant power of the directional light source.

```

function FIND-ELBOW-POINT(SRBF [] expansion, float threshold)
1 float increment ← LUMINANCE(expansion[0])
2 float total ← increment
3 int elbow ← 0
4 while elbow < LENGTH(expansion) ∧ increment/total > threshold do
5   elbow ← elbow + 1
6   increment ← LUMINANCE(expansion[elbow].value)
7   total ← total + increment
8 end while
9 return elbow
    
```

Figure 4.5 Pseudocode for computing the elbow point for the visual importance of the SRBF expansion.

We construct the directional light approximation by bounding the pointwise error of the combined approximation. This approach leads us to a greedy algorithm for selecting the subset of SRBFs which are to be used as directional light sources. The greedy algorithm works as follows. We first sort the SRBFs by decreasing pointwise error in the combined approximation, e.g., the SRBFs are sorted in order of importance in the sense of their luminous power. Constructing the directional light approximation by from the sorted SRBF list, we are in essence bounding the pointwise error of the combined approximation, i.e., $\|E - \tilde{E}_n\|$ decreases monotonously with respect to the number of directional lights n constructed from the sorted SRBF list.

Our final goal is a visually pleasing result while minimizing the rendering costs. To address this we choose the number of directional lights relative to their contribution to the final illumination by applying *Webers law*. Webers law can be interpreted such that the human visual system is unable to distinguish small changes in luminosity. In practice we use a user defined threshold to control the quality of the directional light approximation. We use this threshold as follows. We use the sorted list of SRBFs to iteratively construct the directional lights corresponding to SRBFs until we find that the luminance contribution of the remaining SRBFs falls below the user defined percentage of the total luminance induced by the light sources. This decision can be made by simply considering each SRBF in the list in decreasing order and comparing the luminance of the current SRBF to the accumulated luminance of the previously analyzed SRBFs. Pseudocode for finding the elbow point is shown in Figure 4.5.

We have now obtained the set of centers for visually important directional light sources of the original environment illumination. The next step is to assign radiosity for the directional light sources. As mentioned before, we use the directional light source model to approximate light sources with small solid angle. Thus, we define the radiosity of each light source to be directly proportional to the solid angle of the light source. We use a fixed radius R for each light source and compute the solid angle of the spherical cap $D(\Xi_i, R) = 2\pi(1 - \cos R)$. The radiosity of the directional light source i is then $2\pi(1 - \cos R)\Phi_i$, where Φ_i is the radiant power. We define the radiant power as $\Phi_i = \lambda_i - P(\Xi_i)$, so that the energy is conserved.

4.4 Rendering

Rendering the scene is the last of the three phases of our algorithm and it results in a image of the scene lit by distant environment illumination as seen from a given viewpoint. The rendering is done

4.4 Rendering

```

procedure RENDER-SCENE(SHExpansion  $P$ , DirectionalLightSource []  $lights$ )
1 CLEAR-DEPTH-BUFFER()
2 ENABLE-DEPTH-WRITE()
3 UPDATE-LIGHTING( $mesh, P$ )
4 RENDER-MESH( $mesh$ )
5 DISABLE-DEPTH-WRITE()
6 for each directional light source  $l$  in  $lights$  do
7   ShadowMap  $map \leftarrow$  COMPUTE-SHADOW-MAP( $l.direction$ )
8   RENDER-SHADOW-MAP( $map, l.radiance$ )
9 end for

```

Figure 4.6 Pseudocode for the multipass rendering method. In the first pass we fill the depth buffer and use the precomputed radiosity solution to obtain the per-vertex colors of the mesh (lines 1–4). After the first pass, we reuse the filled depth-buffer and therefore we write only to the color buffer using additional blending. We compute a shadow map for each of the directional light sources l and we use a pixel shader to add the contribution of each directional light source l . These steps are realized by the by the procedures COMPUTE-SHADOW-MAP(Vec3 dir) and RENDER-SHADOW-MAP(ShadowMap map) respectively. (lines 5–9).

in multiple passes. In the first pass, we render the scene using the precomputed radiance transfer solution. We use the low-degree spherical polynomial P , obtained via the decomposition, as the source lighting environment. After the first pass, all rendering is done using additive blending, since we are calculating the sum as in Equation 4.1. We compute a shadow map for each directional light source L_i and use the shadow map to additively blend the lighting contribution of each directional light L_i . Pseudocode for the rendering algorithm is shown in Figure 4.6.

We use percentage-closer-filtering (PCF) to reduce the aliasing artifacts caused by undersampled shadow maps. PCF filtering also enables the smoothing of hard shadow boundaries, which corresponds better with our directional light approximation model, since the solid angle of our directional lights are positive. PCF filtering works by estimating the fractional visibility value $V(x, \omega)$ of a visibility query, where x is the visibility query point and ω is the direction of the visibility query. The fractional visibility query value is computed by averaging the visibility query values in some finite neighborhood $B(x, R)$ of the original visibility query point x . Thus, using PCF we set $V_{PCF}(x, \omega) \approx 1/n \sum_{k=1}^n V(x_i, \omega), x_i \in B(x, R)$. This averaging process thus smooths the aliasing artifacts near hard shadow boundaries. This smoothing effect is evident, if we compare the PCF-method to standard shadow mapping, where the visibility estimate is a binary function $V_{SM} = V(x, \omega)$ and thus highly amenable to the aliasing artifacts caused by undersampled depth fields of the target light source, e.g., too low a shadow map resolution.

In order to map the high-dynamic range output of our precomputed radiosity solution to suitable values for computer displays, we employ a simple form of tone mapping. Tone mapping is the process of transforming arbitrary, i.e., real-valued radiance values to the discrete range of 0 – 255, which is suitable for computer displays. We use a simple exponential model for tone mapping the r, g, b values of the source radiance of the final image. That is, we set $y_i = 1 - e^{-\gamma c_i}, i = 1, 2, 3$, where $y_i, i = 1, 2, 3$ is the final color value and γ is a user-controlled exposure value.

Chapter 5

Experimental Results

In this chapter we present experimental results of the hybrid rendering algorithm for environment illumination. We compare our results against diffuse precomputed radiance transfer using ray-traced images as the reference images. We verify our results using pixelwise image differences between our method, PRT, and the reference images. In addition we demonstrate a visual comparison of the results obtained with our method and we again compare the results against diffuse PRT.

The diffuse PRT solutions for direct lighting were rendering in a way similar to Sloan et al. [66]. Before rendering the diffuse direct lighting, we projected the source lighting environment represented as a cubemap to the spherical harmonic basis in order to obtain the truncated spherical harmonic expansion representing the illumination. We used a similar pipeline for the environment maps for both direct diffuse PRT and our method.

The reference images were computed by ray-tracing. We sampled the environment cubemap exhaustively by supersampling each environment cubemap texel with a 4x4 jittered grid. The resolution of the environment cubemaps we used was 6x64x64. We used one primary ray per image pixel so that the resulting images would be comparable to the ones produced by the real-time algorithms, e.g., no antialiasing scheme should be used in synthesizing the reference images.

5.1 Test Setup

The tests were run on a AMD Athlon 64 3000+ processor system with 1 gigabyte of memory equipped with an NVIDIA GeForce 6600 GT graphics card with 128 megabytes of video memory.

5.1.1 Test Scenes

We used the TEAPOT scene as a simple test case which is able to demonstrate complex shadowing effects due to environment illumination. The scene contains the Utah teapot residing on a small table, as seen in Figure 5.2. The table is included to capture the shadows caused by the teapot.

We use five environment illumination maps, capturing different features characteristic to real-world environment illumination as test cases for our method. The light probes corresponding to the environment illumination functions are show in Figure 5.1.

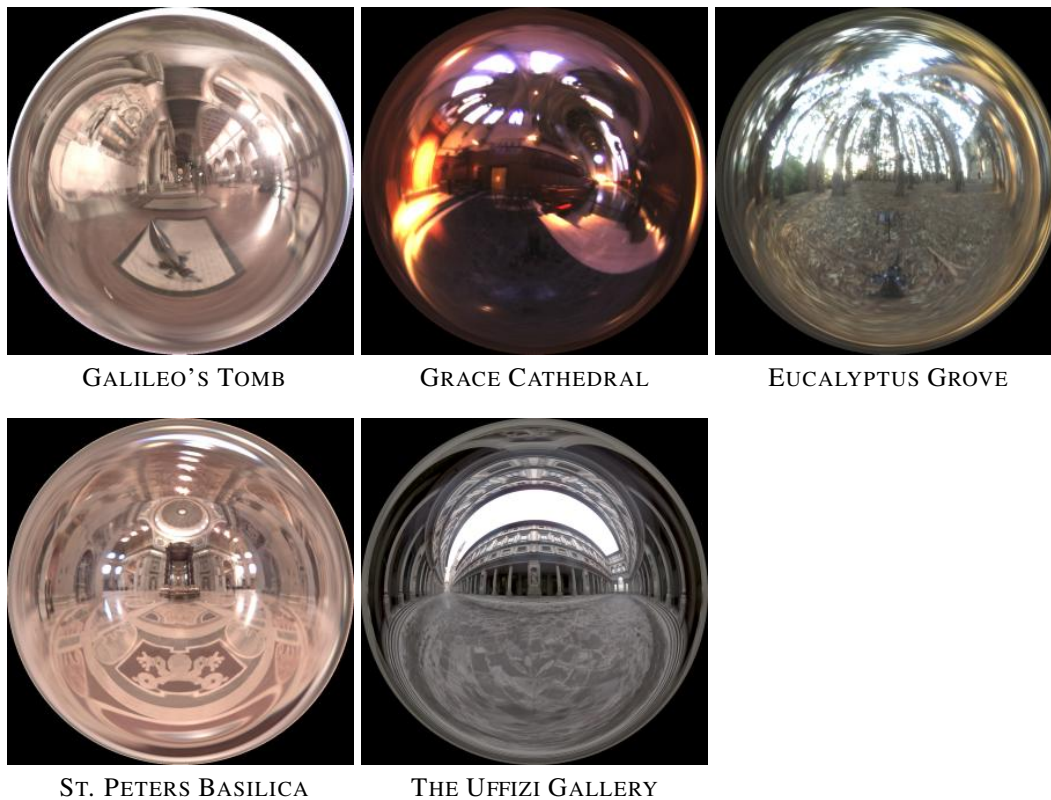


Figure 5.1 HDR light probe images used as test cases for the hybrid rendering algorithm. The light probes shown above demonstrate different features characteristic to real-world environment illumination. The light probe images are courtesy of Paul Debevec [17].

GALILEO'S TOMB is an indoor illumination environment with 7000 : 1 dynamic range, containing windows which act as small, high-intensity light sources.

GRACE CATHEDRAL is an indoor illumination environment with high 200000 : 1 dynamic range. GRACE CATHEDRAL has lots of concentrated, high-intensity features which are not well captured by low-degree spherical polynomials.

EUCALYPTUS GROVE is an outdoor natural illumination environment with 5000 : 1 dynamic range. It features partly clouded sky with high-intensity sun shining through the woods, creating variation in the lighting pattern emerging from the occluded sunlight.

ST. PETER'S BASILICA is an indoor illumination environment with high, 200000 : 1 dynamic range with lots of small concentrated light sources in the ceiling.

THE UFFIZI GALLERY is a natural outdoor illumination environment featuring small 500 : 1 dynamic range and a cloudy, diffuse skylight. The sky is not uniform though, since there is a significant intensity spike caused by the sun which is shining through the clouds.

5.2 Results

In this section we demonstrate experimental results of our method. The ultimate quality of any rendering algorithm is best judged by visually examining the results in comparison with the requirements that were set a priori. In the case of realistic image synthesis the evaluation of an approximative method should be done against a reference solution, since the reference solution is generally obtainable by running a full lighting simulation of the scene. Although visual quality is of paramount importance when evaluating any rendering algorithm one should not rely only on this somewhat subjective evaluation metric. To this end we demonstrate a quantitative comparison of our rendering algorithm against the direct diffuse precomputed radiance transfer as presented by Sloan et al. in 2002 [66]. The pixelwise intensity difference images show that our method captures the shadowed regions near the occluder more accurately compared to PRT. On the other hand, as is evident from the error visualizations our method also introduces error to the resulting images. This fact is further verified by simple error statistics drawn from the difference images and we will discuss these matters later in this section.

5.2.1 Visual Comparison

A visual comparison of our method against the ray-traced reference images is shown in Figures 5.2 - 5.6. The alert reader should focus on regions near the teapot, i.e., regions which are highly occluded by the teapot geometry. In these regions one should look for subtle shadowing effects and differences between our method and the ground truth.

In Figure 5.2 the TEAPOT scene is lit by the GALILEO'S TOMB lighting environment. The environment illumination is dominated by the sunlit windows. This causes the sharp shadows on the table behind the teapot. The shapes of the shadows are similar in both images, although the quality of the shape approximation decreases as the distance to the occluding geometry increases in the image to the right. This is a direct result of the directional light approximation. Another artifact of the directional light model is found on the teapot, which is brighter than it should be.

In Figure 5.3 the TEAPOT scene is lit by the GRACE CATHEDRAL environment illumination map. It is worth noticing that the main shadow features are similar in both images. The reference solution contains more fine-grained shadows, which is to be expected of course. One other thing of interest is the orange highlight on the teapot. This is caused by a concentrated high-intensity light source in the original environment illumination. The original feature is not captured well by a low-degree spherical polynomial representing the original environment map, but our method is able to amplify this effect thus increasing the quality of the approximation.

In Figure 5.4 the TEAPOT scene is lit by the EUCALYPTUS GROVE environment illumination map. In this case the main shadow features are again similar with the difference that our method exaggerates the long shadows caused by the sun shining through the woods on the table. This overestimation grows worse as the distance to the occluder, i.e., teapot increases. The overestimation of shadows is an artifact of the directional light model used to approximate the light sources.

In Figure 5.5 the TEAPOT scene is lit by the ST. PETER'S BASILICA environment illumination map. The original illumination environment contains many isolated, high-intensity light sources. This kind of lighting environment should be an ideal target for our hybrid rendering algorithm. From the side-by-side comparison we can notice that our method produces visually pleasing approximations to the physically correct sharp shadows due to the localized, high-intensity light sources found in the original environment illumination map.

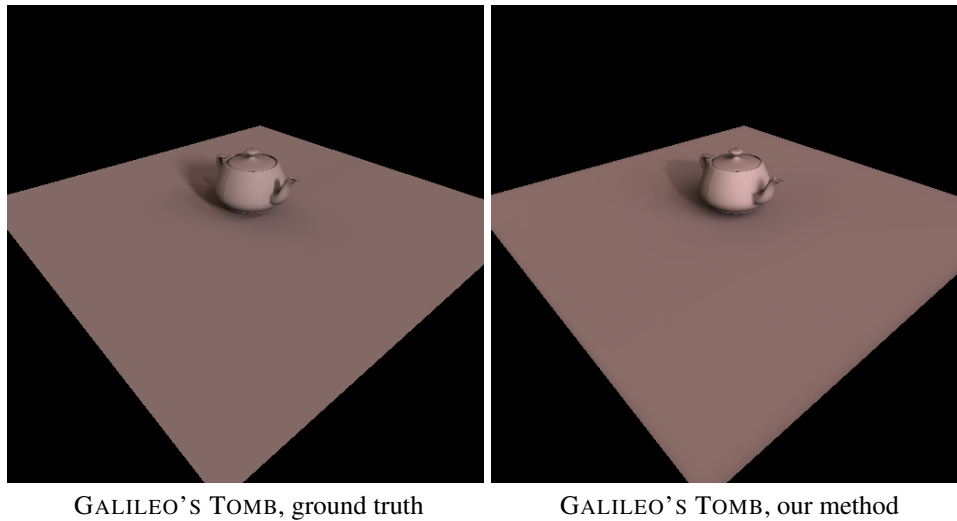


Figure 5.2 Visual comparison of the results. The TEAPOT scene is lit by the GALILEO'S TOMB environment illumination map. The GALILEO'S TOMB environment illumination map contains a high-intensity, concentrated light source which causes the hard shadow behind the teapot.

Our last test case is the UFFIZI GALLERY. A side-by-side comparison featuring the TEAPOT scene is shown in Figure 5.6. The features introduced by our method are subtle but existent. Our method amplifies the definition of the shadow of the teapot's nozzle on the table. Admittedly, the results of our algorithm almost reduce to those obtained with diffuse PRT only. This behavior is to be expected as well, since the original lighting environment is slowly varying as a result of the cloudy sky and uniform reflections from the diffuse materials in the buildings. We conclude that our algorithm behaves as it is supposed to, i.e., resort to PRT as much as is effective.

5.2.2 Quantitative Comparison

In this section we compare our method to diffuse precomputed radiance transfer [66]. We compare the quantitative image quality with respect to the reference images. The memory costs of these two methods are essentially the same, since our method augments the PRT method by utilizing hardware-supported directional lights. This adds a small overhead to the overall performance of our method in favor of the PRT. Nevertheless, both of these methods are suitable for real-time rendering of moderately complex scenes consisting of hundreds of thousands of triangles.

We use the same test scenes and lighting environments as in the previous section. The results are shown in Figures 5.7 - 5.11. Images in the upper row in these figures are obtained by using our method and diffuse precomputed radiance transfer solution respectively. Images in the lower row display absolute pixelwise intensity differences with respect to the reference image.

In Figure 5.7 the TEAPOT scene is lit by GALILEO'S TOMB environment illumination map. The most striking difference of our method compared to PRT is the sharp shadow on the table behind the teapot. As seen from the error visualizations, our method exhibits smaller error in that region. On the other hand, our method produces greater error in the front of the teapot. This is explained by the high-intensity directional light causing the sharp shadow behind the teapot, which is used to approximate the concentrated light source in the original environment illumination map. At the

5.2 Results

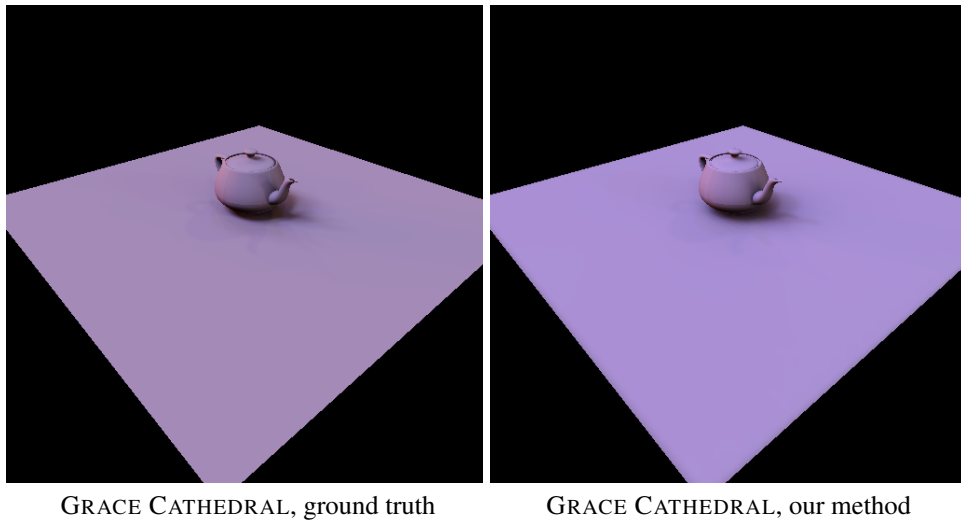


Figure 5.3 Visual comparison of the results. The TEAPOT scene is lit by the GRACE CATHEDRAL environment illumination map. Notice the subtle shadowing effects on the table near the teapot. These are caused by ceiling windows, which act as small, high-intensity light sources in the scene. Worth noticing is also the amplified orange highlight on the teapot.

front of the teapot, the directional cosine falloff term of the directional light approximation is large and thus leads to overshooting the illumination in that region. This test case demonstrates that our method trades the accuracy in shadow reproduction to error in the unoccluded lighting.

In Figure 5.8 the TEAPOT scene is lit by GRACE CATHEDRAL environment illumination map. In this case there are two things worth noticing. First, as seen from the error visualization our method improves the shadow approximation on the table near the teapot. Second, our method is able to reproduce the orange highlight found on the teapot. This fact is also verified by the error visualization; the reddish spot on the teapot found in the PRT error image is mostly missing from error image resulting from our method. It should also be noticed that our method increases the error in regions front of the teapot, again, resulting from the directional light approximation.

In Figure 5.9 the TEAPOT scene is lit by EUCALYPTUS GROVE environment illumination map. The patterns in the error visualizations have similar form, although the patterns appear smoother in the case of PRT. This test case demonstrates again, that our method improves the approximation to the lighting near the teapot, as seen from the error images by examining the spot on the table behind the teapot.

In Figure 5.10 the TEAPOT scene is lit by ST. PETER'S BASILICA environment illumination map. As seen from the error images, our method performs consistently better in comparison to the PRT-method. This is mostly explained by the fact, that the ST. PETER'S BASILICA contains lots of isolated, high-intensity and spatially concentrated light sources. This renders it very suitable for our method.

A somewhat counterexample to the ST. PETER'S BASILICA environment is the THE UFFIZI GALLERY. The former contains many concentrated light sources which are isolated whereas the latter is mostly diffuse, i.e., slowly varying. Figure 5.11 shows the TEAPOT scene lit by the UFFIZI GALLERY environment illumination. The overall impression is that our method demonstrates greater error. As before, our method improves the approximation near the occluding geometry, but as

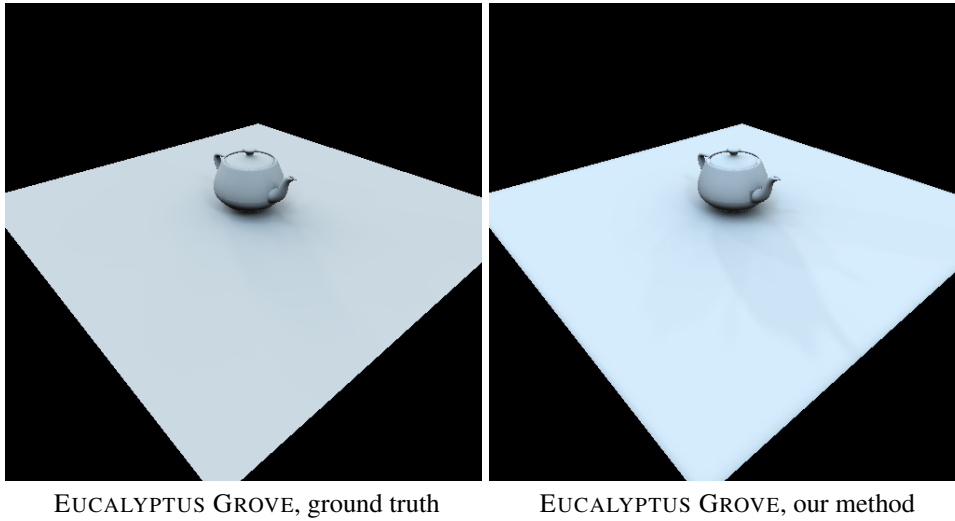


Figure 5.4 Visual comparison of the results. The TEAPOT scene is lit by the EUCALYPTUS GROVE environment illumination map. The long shadows on the table are due to the sun shining through the woods. It is worth noticing, that our approximative method compares well near the occluder, i.e., near the teapot. On the other hand, the long shadows look unrealistic as the distance to the occluder increases. This is expected, as more lighting from the environment enters the hemisphere in points far away from the occluder, since the visible area of the hemisphere grows.

a result it introduces additional error on the table in front of the teapot. The overall error is explained by the fact, that the sun shines through the cloudy sky in the UFFIZI GALLERY environment, and thus it creates an intensity spike. Now, our method approximates this intensity spike with directional light and this approximation error is almost constant over the plane.

These quantitative results demonstrate that our method improves the accuracy of the approximation near occluding geometry. This improvement does not come for free, since as we have seen, the point light model used in our method causes error further away from the occluding geometry as well as visible parts of the scene. This is to say that our method trades shadows for some other form of error. From this point of view it seems that there are no absolute quantitative results that encourage the wide application of our method in all environment illumination configurations. It should be noted though, that visually pleasing shadows might justify the use of our method in most cases. Furthermore, we have yet to found a test case in which our method produces unacceptable results. This is to say that although the performance of our algorithm is dependent on the original environment illumination it is applied, it does not produce substantially worse results in any case than the diffuse PRT only. To summarize, in the light of these results, our algorithm succeeds in augmenting the diffuse PRT-method by increasing the accuracy of shadow reproduction, trading shadow accuracy for error in regions with high visibility. The source for the increase in the image error lies in the directional light model used to represent the decomposed high-intensity light sources. It is important to note that, in the light of these results, the directional light model provides poor approximation to the area light sources found by the decomposition process. We will discuss this matter further in the Chapter 6.

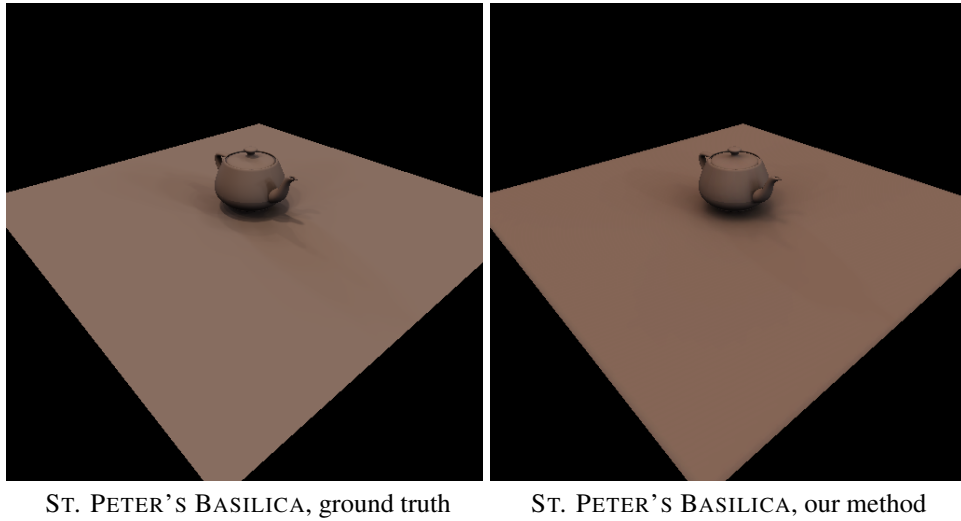


Figure 5.5 Visual comparison of the results. The TEAPOT scene is lit by the ST. PETER'S BASILICA environment illumination map. Notice the sharp shadows on the table due to concentrated light sources found in the ceiling of the lighting environment. Our proposed rendering algorithm provides visually pleasing, although not physically correct approximations of these features.

Error Statistics

As we presented the quantitative results of our method compared against diffuse PRT, we conjectured from the error visualizations that our method spatially transforms error near the occluding geometry to error further from the occluding geometry or error on the unoccluded geometry. To somehow verify this speculation we computed statistics from the error images. The mean and standard deviation of the error over the images are shown in Table 5.1. From the statistics we can conclude that our method performs worse on the average than diffuse PRT on all the scenes except ST. PETER'S BASILICA. GALILEO'S TOMB is somewhat similar illumination environment as ST. PETER'S BASILICA which is verified also by the fact that the mean errors in that test case are approximately the same. These statistics reveal that the performance of our method is highly dependent of the lighting environment. All in all we can conclude that our method increases the approximation quality in shadowed regions near the occluding geometry and decreases the approximation quality in the unshadowed regions with the penalty of increased average error. The increase in the average error is not a severe limitation, since it is well known that the human eye is more sensitive to discontinuities in the lighting than to a change in the average intensity.

5.2.3 Additional Results

We have so far run our algorithm on a relatively simple test scene. In this section we demonstrate additional results featuring a moderately complex scene consisting of hundreds of thousands of triangles.

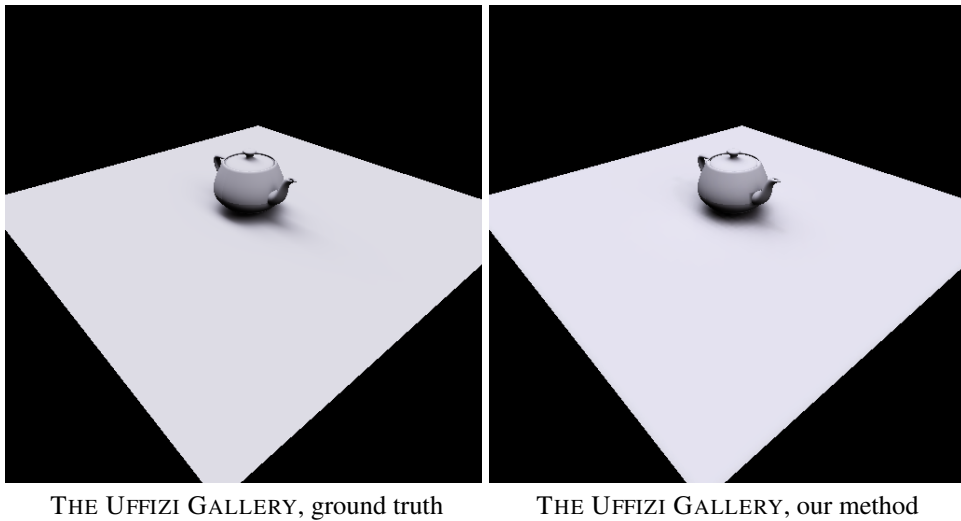


Figure 5.6 Visual comparison of the results. The TEAPOT scene is lit by the UFFIZI GALLERY lighting environment. The lighting environment features diffuse skylight with intensity spikes in the region of sun shining behind the cloud cover. Therefore the lighting environment can be described as slowly varying and our method uses only few point lights to amplify the effects caused by the cloud-covered sun. This can be seen from the sharp shadow of the teapots nozzle on the table. This feature is missing from the direct diffuse PRT solution.

scene	method	mean error	standard deviation
GALILEO'S TOMB	PRT	0.0065	0.0098
GALILEO'S TOMB	our method	0.0068	0.0124
GRACE CATHEDRAL	PRT	0.0141	0.0191
GRACE CATHEDRAL	our method	0.0236	0.0391
EUCALYPTUS GROVE	PRT	0.0258	0.0307
EUCALYPTUS GROVE	our method	0.0307	0.0395
ST. PETER'S BASILICA	PRT	0.0271	0.0310
ST. PETER'S BASILICA	our method	0.0128	0.0173
THE UFFIZI GALLERY	PRT	0.0092	0.0118
THE UFFIZI GALLERY	our method	0.0143	0.0178

Table 5.1 Error statistics of the absolute difference images. We computed the absolute pixelwise luminous intensity difference in comparison to the ground truth image. We then computed the mean and standard deviation of per-pixel error, which are shown above for each lighting environment.

5.3 Strengths and Weaknesses of the Hybrid Rendering Algorithm

In this section we analyze the properties of the proposed rendering method. We base our analysis on the experimental results demonstrating the performance of the algorithm in a few well chosen example cases.

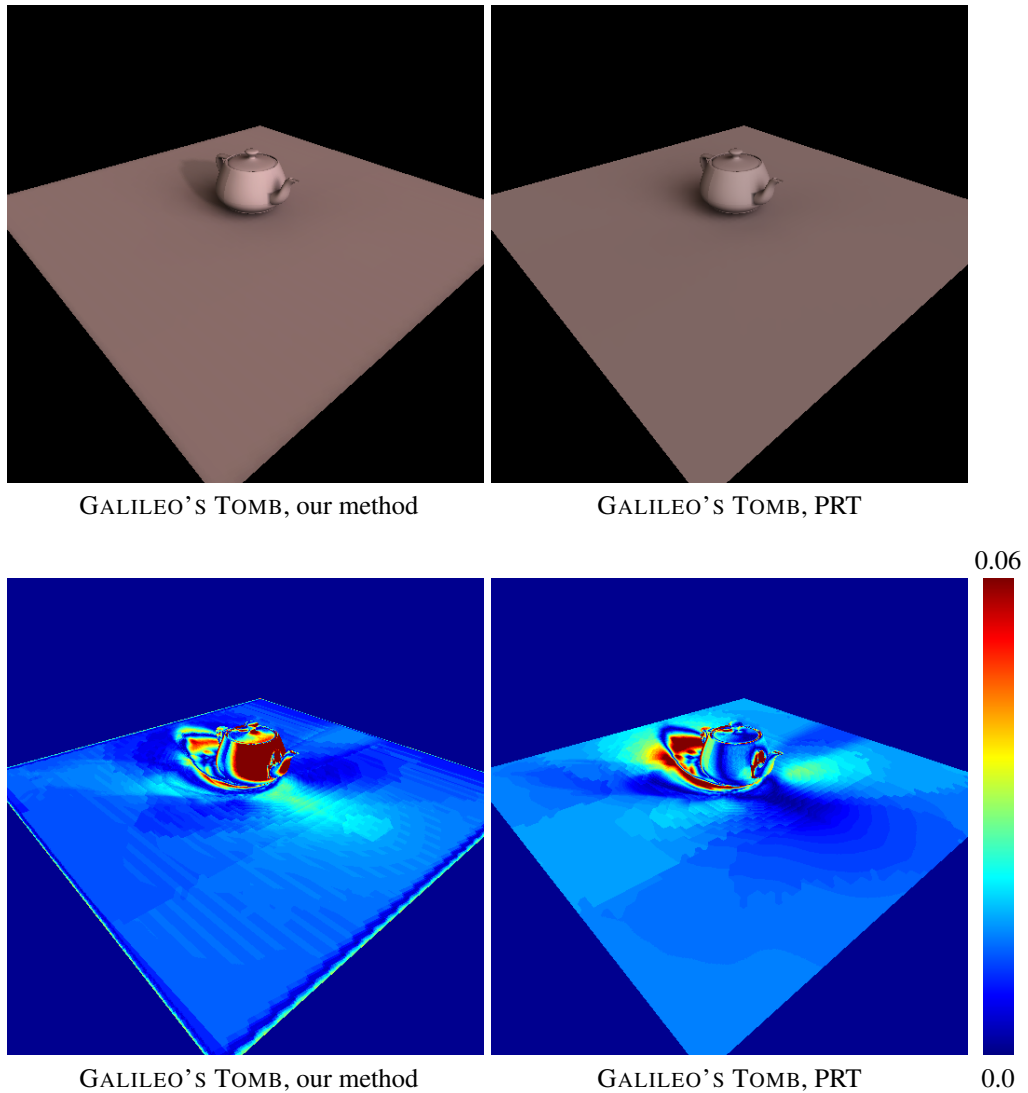


Figure 5.7 Quantitative comparison of the results. The TEAPOT scene is lit by GALILEO'S TOMB environment illumination map. Images on the upper row are obtained by using our method and direct diffuse precomputed radiance transfer solution respectively. Images on the lower row display absolute pixelwise intensity differences with respect to the reference image.

5.3.1 Strengths of the Algorithm

One of the greatest strengths of our algorithm is its simplicity. Quite unexpectedly, various other, generally regarded as more advanced or robust methods failed to produce adequate results for the point light approximation. We experimented with constrained and unconstrained least-squares solutions together with standard multiresolution analysis using the Haar-system over the cubemap faces only to finally found out that the simple approach described in this thesis worked at least equally well in all cases.

Second, one major limitation for spherical harmonic based precomputed radiance transfer is the

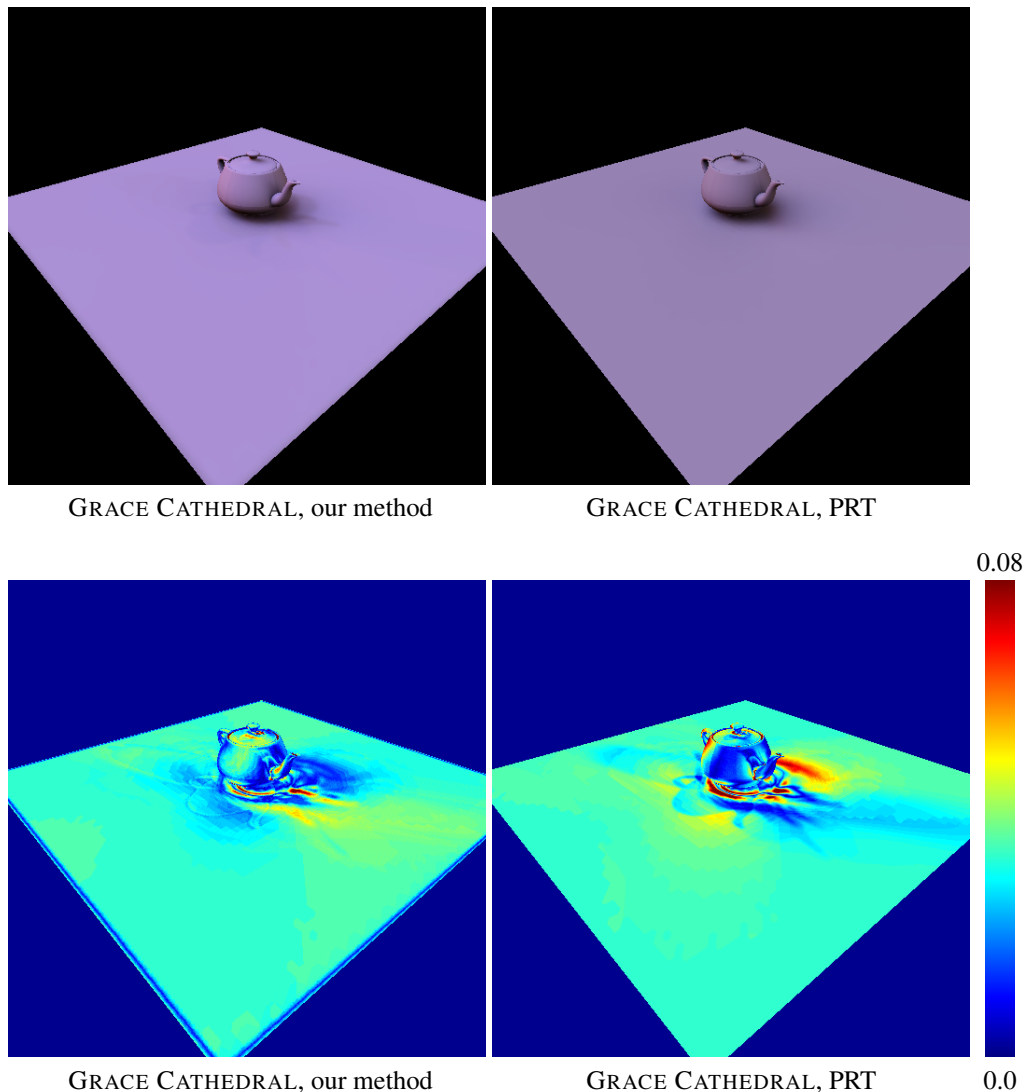


Figure 5.8 Quantitative comparison of the results. The TEAPOT scene is lit by GRACE CATHEDRAL environment illumination map. Images on the upper row are obtained by using our method and direct diffuse precomputed radiance transfer solution respectively. Images on the lower row display absolute pixelwise intensity differences with respect to the reference image.

inability to produce sharp shadows unless resorting to supersampling the transfer functions, e.g., increasing the density of the triangulation of the scene. Our method is able to render sharp shadows due to concentrated, high-intensity light sources in the original, i.e., unapproximated environment illumination. The quality of these shadows is dependent on the chosen real-time shadowing technique, and therefore somewhat independent of the scene geometry. That is, we do not have to modify the scene geometry, e.g., by subdividing the triangles to increase the visibility sampling resolution. For example, by using shadow maps with adequate resolution, we are able to sample visibility with arbitrary ¹ degree inside one triangle. This is in contrast to current precomputed radiance transfer methods, in which the visibility is sampled over triangle vertices of the scene. All this boils down to

¹With respect to the limitations of the rendering hardware, of course.

5.3 Strengths and Weaknesses of the Hybrid Rendering Algorithm

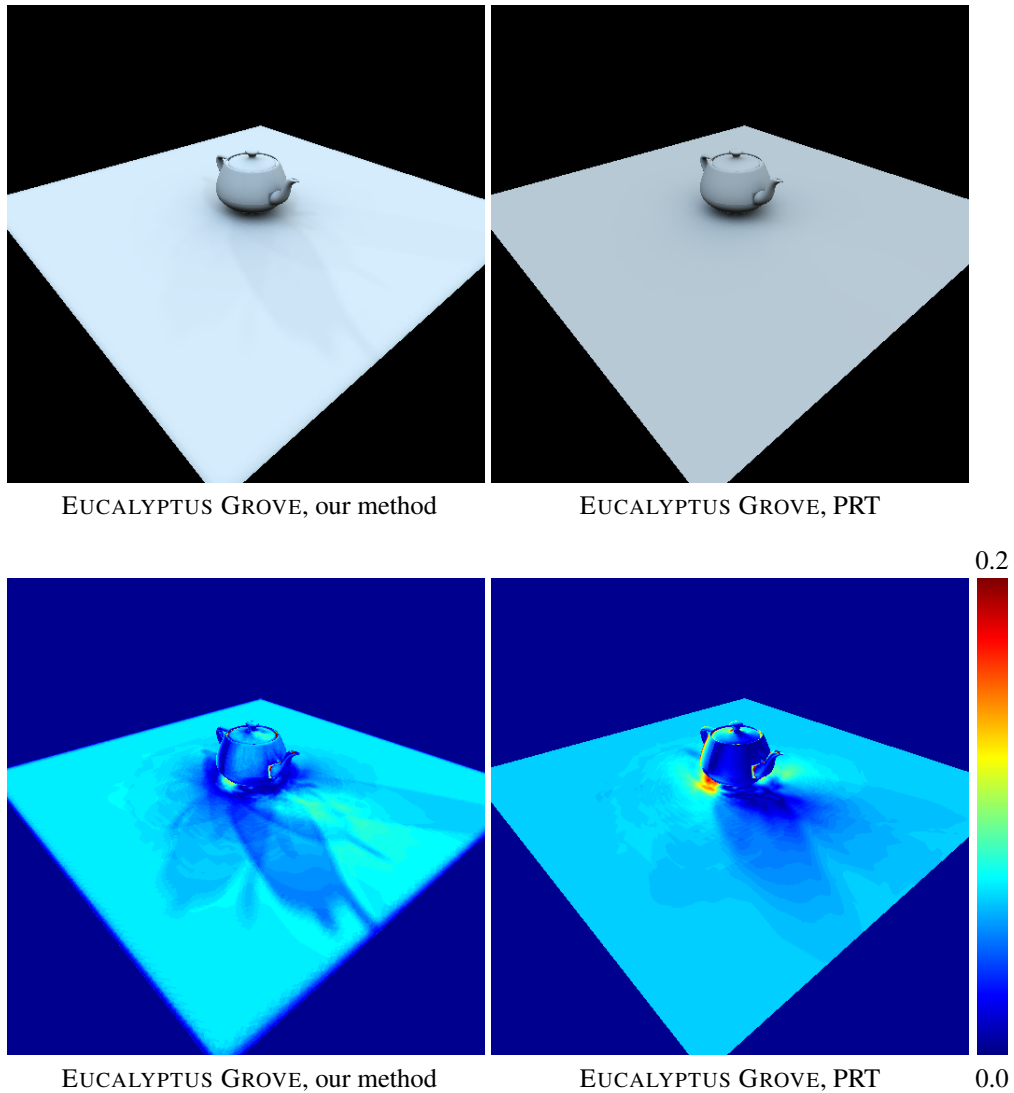


Figure 5.9 Quantitative comparison of the results. The TEAPOT scene is lit by EUCALYPTUS GROVE environment illumination map. Images on the upper row are obtained by using our method and direct diffuse precomputed radiance transfer solution respectively. Images on the lower row display absolute pixelwise intensity differences with respect to the reference image.

the fact that we are able to reproduce sharp shadows with relatively low overhead by utilizing state-of-the-art real-time shadowing techniques and advances in consumer-level rendering hardware.

Third, we expect that our method exhibits reasonable temporal coherence since we work with smooth spherical functions and therefore our algorithm works reasonably well with time-continuous environment illumination. Effects such as switching lights on and off in an otherwise dark room are discontinuous by nature and will cause rapid changes in the lighting in any case. Time-continuous effects include continuous animation of the light position and color for instance. Our algorithm does not suffer from excessive flickering in time-continuous dynamic environment illumination, although the truncation of the point light approximation series might cause flickering artifacts. In

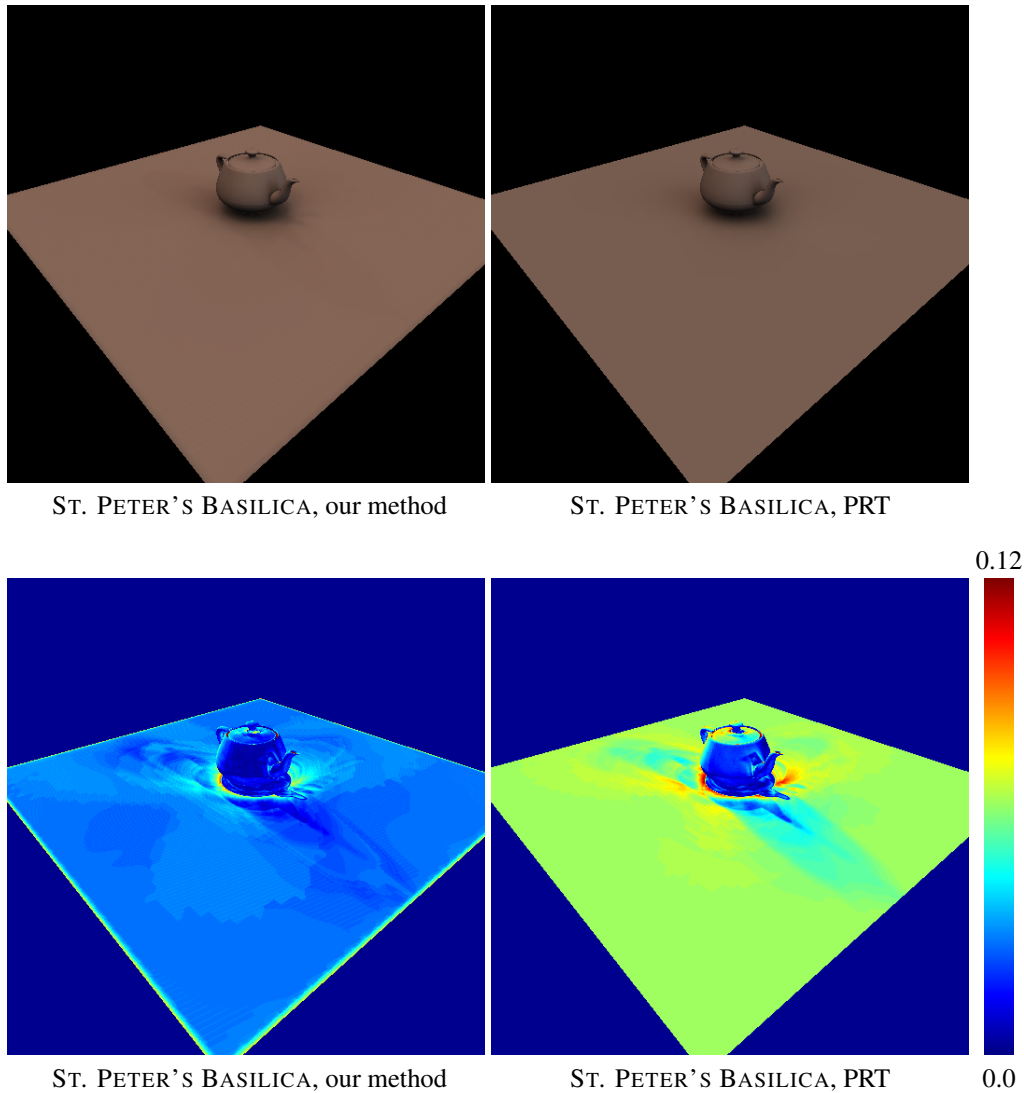


Figure 5.10 Quantitative comparison of the results. The TEAPOT scene is lit by ST. PETER'S BASILICA environment illumination map. Images on the upper row are obtained by using our method and direct diffuse precomputed radiance transfer solution respectively. Images on the lower row display absolute pixelwise intensity differences with respect to the reference image.

addition to the truncation, too small a sampling rate may introduce aliasing artifacts to the point light approximation and thus cause flickering effects in the rendered images. Our method is fully deterministic and therefore it does not suffer from the same problems related to temporal coherence as point light approximation methods based on stochastic sampling, namely the randomness of the generated sample locations between consecutive animation frames. Therefore we expect that our method exhibits better temporal coherence in the generated point light sets than methods based in stochastic sampling.

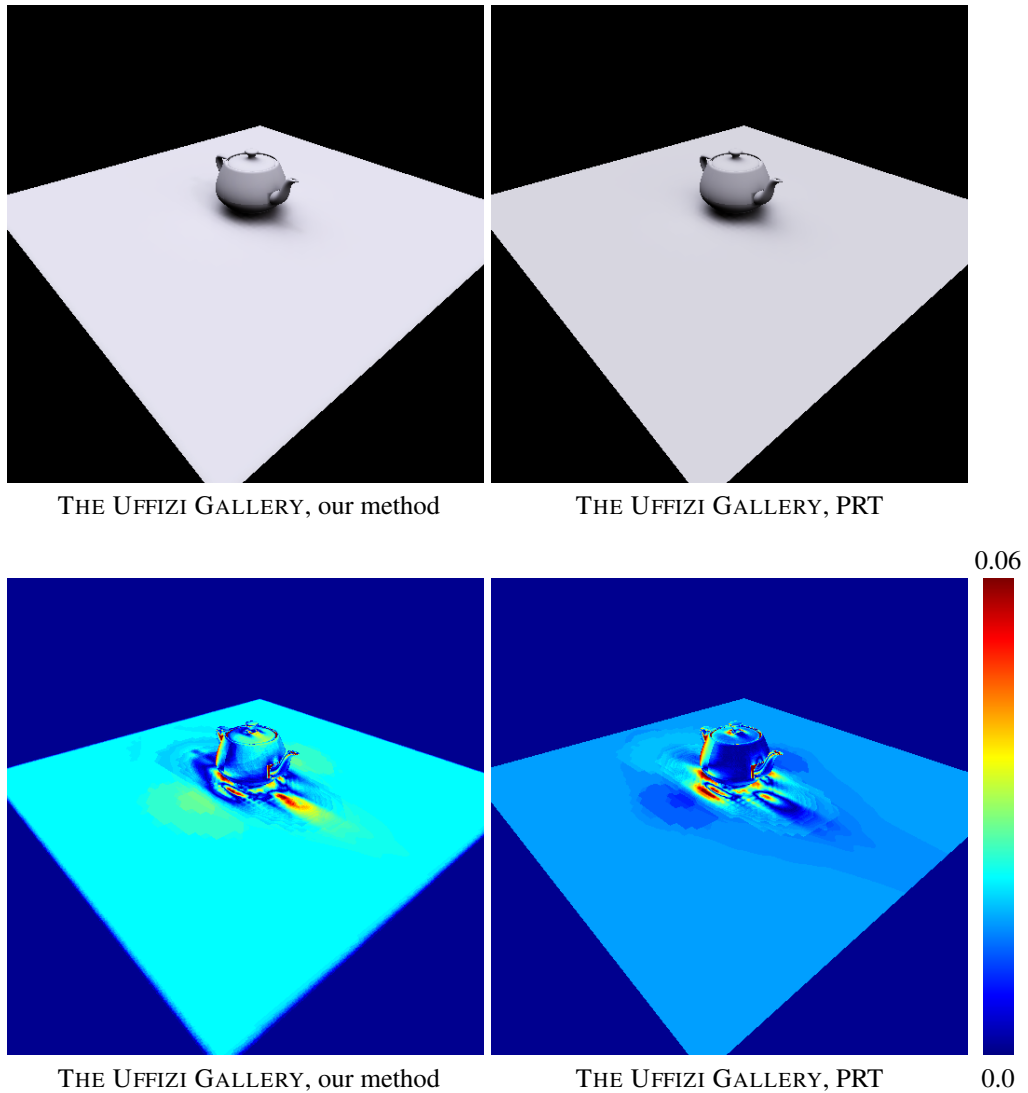


Figure 5.11 Quantitative comparison of the results. The TEAPOT scene is lit by THE UFFIZI GALLERY environment illumination map. Images on the upper row are obtained by using our method and direct diffuse precomputed radiance transfer solution respectively. Images on the lower row display absolute pixelwise intensity differences with respect to the reference image.

5.3.2 Weaknesses of the Algorithm

Although our algorithm is capable of synthesizing sharp shadows originating from concentrated light sources, there is some approximation error due to the point light model used to produce the shadows. In other words, the current method is not physically correct. Artifacts due to point light approximation are easily seen in scenes in which the light source causes long shadows relative to the shadow caster, e.g., the distance between the shadow and the occluding geometry is relatively large. This error grows as the distance between the occluding and the receiving geometry increases. This is to be expected, since as the distance to the occluder grows, the occluded solid angle, as seen from the receiving geometry, decreases so that more light enters from the environment to the receiving point



THE ST. PETERS BASILICA, our method



THE ST. PETERS BASILICA, PRT



THE ST. PETERS BASILICA, ground truth

Figure 5.12 Visual comparison of the results. The DRAGON scene is lit by the ST. PETERS BASILICA lighting environment. As seen from the images, our method is able to capture the coarse features of the detailed shadowing effects caused by point-like light sources present in the lighting environment. On the other hand, our method shows some overall difference in the color hue. Compared to PRT, our method is able to reproduce visually pleasing sharp shadows considerably better.

thus softening the shadow due to the decreased occlusion of environment illumination. Artifacts due to point light approximation could be overcome by using a real-time soft-shadow algorithm which is able to approximate shadows from area light sources. To overcome the limitations due to the used point light model, we could integrate a real-time soft-shadow algorithm, such as [8] to our method with very little modifications to the algorithm itself.

Perhaps the most drastic weakness of the proposed rendering algorithm is that it is not effective for all types of environment illumination. It is assumed that the environment illumination contains concentrated, high-intensity light sources that are not adequately well captured by a truncated spherical harmonic series. This is the case with natural, real-world illumination as showed by Dror et al. [21].

5.3 Strengths and Weaknesses of the Hybrid Rendering Algorithm

There is no advantage in applying our method to slowly varying environment illumination since it has been shown [58] that the truncated spherical harmonic series is an effective approximation tool in that case. It should be noted though, that our method does not fail in these cases either; the decomposition would consist simply of the spherical polynomial part. On the other extreme the environment illumination could be filled with small, concentrated, high-intensity light sources. In that case, our method would approximate some of the overall contribution of this lighting configuration with a low-degree spherical polynomial. This might cause ringing artifacts which are manifestations of too small an approximation degree with respect to the frequency content of the target signal, i.e., the environment illumination function. On the other hand, our method would use a roughly equivalent amount of point lights to approximate those concentrated light sources as there are in the original, unapproximated environment illumination. This might lead to excessive amount of point lights to be rendered in real-time with current consumer level-hardware. Even though our method does not provide physically correct nor even visually indistinguishable images compared to the reference solutions, it seems to be the case that our method succeeds in increasing the quality of the lighting approximation compared to using only the precomputed radiosity solution in computing the approximation. We are able to demonstrate increased image quality with respect to shadow reproduction with small overhead compared to other methods which are capable of rendering sharp shadows from all-frequency environment illumination, such as [53, 74, 78, 49]. In addition, our method is simple and does not involve significant overhead to be used in computer graphics applications. Although our method is not applicable in all types of environment illumination the produced results have proven to be passable approximation none the less.

Chapter 6

Discussion and Future Work

We conclude this thesis by discussion of the presented hybrid algorithm for environment illumination rendering. We also consider some directions for future work, spawning from the ideas presented in this thesis. First, we think that the method presented here could be improved in several ways, increasing the efficiency and the accuracy of the rendering algorithm. Second, we think that the decomposition itself might be used in other interesting application domains, such as importance sampling and BRDF approximation.

6.1 Discussion

Our decomposition can also be seen as a special case of a more general multiresolution analysis techniques, i.e., wavelets [70, 71]. Our method was designed with the characteristic features of the target function in mind. This is in contrast to the more general multiresolution spaces. Although we used a two-level decomposition model it would be possible to generalize the decomposition using a multilevel decomposition space $M_1 \oplus M_2 \oplus \dots \oplus M_n$. The challenges would be, in addition to selecting the appropriate component spaces M_i , to ensure some sort of continuity in the construction of the approximation to the target function, since the spaces M_i do not generally form a nested sequence. This implies the possibility of $M_i \cap M_{i+1} = \emptyset$, which is problematic in the sense of continuity in the approximant, e.g., the approximation might be constructed using only one component space at a time. This is not the case when applying standard multiresolution analysis with nested spaces $M_1 \subset M_2 \subset \dots \subset M_n$. Therefore we conclude that generalizing the decomposition model is not an easy task in practice, since it requires considerable effort in designing the component spaces and the decomposition method.

6.2 Conclusions and Future Work

In future work we are hoping to cover to a couple of different research directions. First, there are several improvements to the algorithm that we have not yet tried. Second, we believe that the decomposition model itself, although it is really simple, might prove itself useful in some other instances of spherical function analysis and approximation.

Improving the Algorithm

We would like to improve the efficiency as well as the accuracy of our method. One approach could be to clusterize the receiver triangles by similar surface normals. We would then select a set of n directions Ψ_i corresponding to the clustered normals and then apply the decomposition procedure in the induced hemispheres $\Omega(\Psi_i)$. This should lead to increased efficiency in cases where the normals are not distributed evenly over the sphere, scenes with a floor for instance. In addition we might see an increase in the image quality as well, since by culling certain useless lights from the approximation we are free to use more lights within the same point light budget.

The point light approximation fails in cases where the distance from the occluding geometry between the point light source and the shadow receiving surface grows, as we use the point light model to approximate light sources with positive solid angle, i.e., area light sources. The key factor here is to somehow approximate the lighting from the area light sources. One option would be to use a sophisticated soft-shadow algorithm designed for area light sources for that purpose. It might also be of moderate interest to experiment with somewhat more simple techniques that might lessen the artifacts caused by the point light approximation.

One particularly simple method would be to deduce the shadow size from the size of the light source. The shadow size would be controlled by increasing or decreasing the size of the PCF reconstruction kernel with respect to the size of the light source. Other possibility would be to alpha blend the hard shadows by computing an occlusion factor for the vertices of the scene. This technique relies on the observation that the more a surface point is occluded the less there is incoming light to that point and hence the point should be more shadowed. The reverse conclusion holds for soft shadows, since small occlusion allows more environment illumination to contribute to the lighting at the surface point softening the shadow due to an occluded point light source. The occlusion factor could be precomputed along with the radiosity solution, or it could be obtained by analyzing the spherical harmonic expansions of the precomputed transfer functions, since it is known that occluded directions manifest themselves as zeros in the transfer functions.

Conclusions

In this thesis we have proposed a novel hybrid rendering algorithm for realistic rendering of static scenes under distant environment illumination. We have shown how to combine real-time shadowing techniques with precomputation-based techniques via spherical function decomposition method.

Our proposed method combines the strengths of the previous precomputation-based methods [66, 53], while avoiding their most significant pitfalls, resulting in an approximative real-time rendering algorithm for static scenes.

We have demonstrated increased image quality with respect to shadow reproduction with small overhead compared to other methods which are capable of rendering sharp shadows from all-frequency environment illumination, such as [53, 74, 78, 49]. We have pointed out the limitations of our method and argued that despite the limitations, our method is able to produce a visually pleasing approximation to the physically correct lighting caused by real-world environment illumination.

Bibliography

- [1] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph.*, 22(3):605–612, 2003.
- [2] Timo Aila. Alias-free shadow maps. In *Proceedings of the 2nd EG Symposium on Rendering*, Springer Computer Science. Eurographics, Eurographics Association, 2004.
- [3] Timo Aila and Tomas Akenine-Möller. A hierarchical shadow volume algorithm. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*. ACM Press, 2004.
- [4] Tomas Akenine-Moeller, Eric Chan, Wolfgang Heidrich, Jan Kautz, Mark Kilgard, and Marc Stamminger. Real-time shadowing techniques. In *GRAPH '04: Proceedings of the conference on SIGGRAPH 2004 course notes*, page 27, New York, NY, USA, 2004. ACM Press.
- [5] James Arvo. The Role of Functional Analysis in Global Illumination. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95*. Springer-Verlag, 1995.
- [6] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In *Proceedings of ACM SIGGRAPH 94*, pages 75–84. ACM Press, 1994.
- [7] Michael Ashikhmin and Peter Shirley. Steerable illumination textures. *ACM Trans. Graph.*, 21(1):1–19, 2002.
- [8] Ulf Assarsson and Tomas Akenine-Moeller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.*, 22(3):511–520, 2003.
- [9] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. *Commun. ACM*, 19(10):542–547, 1976.
- [10] Stefan Brabec and Hans-Peter Seidel. Hardware-accelerated rendering of antialiased shadows with shadow maps. In *Proceedings of Computer Graphics International '01*, pages 209–214. IEEE, 2001.
- [11] Stefan Brabec and Hans-Peter Seidel. Shadow volumes on programmable graphics hardware. *Computer Graphics Forum (Proceedings of Eurographics '03)*, 25(3), 2003.
- [12] Martin D. Buhmann and M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.
- [13] Brian Cabral, Marc Olano, and Philip Nemecek. Reflection space image based rendering. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 165–170, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

-
- [14] Franklin C. Crow. Shadow algorithms for computer graphics. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 242–248, New York, NY, USA, 1977. ACM Press.
- [15] Stephan Dahlke, Wolfgang Dahmen, Ilona Weinreich, and Eberhard Schmitt. Multiresolution analysis and wavelets on s_2 and s_3 , 1994.
- [16] 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 ACM SIGGRAPH 98*, pages 189–198, Orlando, Florida, July 1998. ACM Press.
- [17] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [18] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.
- [19] Yoshinori Dobashi, Kazufumi Kaneda, Hideki Nakatani, and Hideo Yamashita. A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum*, 14(3):229–240, 1995.
- [20] Julie O'B. Dorsey, Francois X. Sillion, and Donald P. Greenberg. Design and simulation of opera lighting and projection effects. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 41–50, New York, NY, USA, 1991. ACM Press.
- [21] Ron O. Dror, Thomas K. Leung, Edward H. Adelson, and Alan S. Willsky. Statistics of real-world illumination. In *CVPR (2)*, pages 164–171. IEEE Computer Society, 2001.
- [22] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. AK Peters, 2003.
- [23] C. Everitt and M. Kilgard. Practical and robust stenciled shadow volumes for hardware-accelerated rendering, 2002.
- [24] György Fekete and Lloyd A. Treinish. Sphere quadtrees: a new data structure to support the visualization of spherically distributed data. In Edward J. Farrell, editor, *Extracting Meaning from Complex Data: Processing, Display, Interaction*, volume 1259/1, pages 242–253. SPIE, 1990.
- [25] György Fekete. Rendering and managing spherical data with sphere quadtrees. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 176–186, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.
- [26] Randima Fernando, Sebastien Fernandez, and Kavita Bala. Adaptive shadow maps. In *Proceedings of SIGGRAPH '01, Computer Graphics Proceedings, Annual Conference Series*, pages 387–390. ACM SIGGRAPH, 2001.
- [27] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1994.
- [28] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM Press.

BIBLIOGRAPHY

- [29] Paul Green, Jan Kautz, Wojciech Matusik, and Fredo Durand. View-dependent precomputed light transport using nonlinear gaussian function approximations. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 7–14, New York, NY, USA, 2006. ACM Press.
- [30] Ned Greene. Environment mapping and other applications of world projections. *IEEE Comput. Graph. Appl.*, 6(11):21–29, 1986.
- [31] Roy Hall. *Illumination and color in computer generated imagery*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.
- [32] Xuejun Hao and Amitabh Varshney. Real-time rendering of translucent meshes. In *ACM Transactions on Graphics*, volume 23(2), pages 120–142. ACM press, 2004.
- [33] Wolfgang Heidrich and Hans-Peter Seidel. View-independent environment maps. In *HWWS '98: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 39–ff., New York, NY, USA, 1998. ACM Press.
- [34] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, hardware-accelerated shading and lighting. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 171–178, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [35] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, New York, NY, USA, 2001. ACM Press.
- [36] James T. Kajiya. The Rendering Equation. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, pages 143–150. ACM Press, 1986.
- [37] Jan Kautz and Michael D. McCool. Approximation of Glossy Reflection with Prefiltered Environment Maps. In *Proceedings Graphics Interface 2000*, pages 119–126, May 2000.
- [38] Jan Kautz, Peter-Pike Sloan, and John Snyder. Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics. In *13th Eurographics Workshop on Rendering*, pages 301–308, June 2002.
- [39] Jan Kautz, Pere-Pau Vazquez, Wolfgang Heidrich, and Hans-Peter Seidel. Unified approach to prefiltered environment maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 185–196, London, UK, 2000. Springer-Verlag.
- [40] Thomas Kollig and Alexander Keller. Efficient illumination by high dynamic range images. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 45–50, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [41] Janne Kontkanen, Emmanuel Turquin, Nicolas Holzschuch, and François Sillion. Wavelet radiance transport for interactive indirect lighting. In Thomas Akenine-Möller Wolfgang Heidrich, editor, *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*. Eurographics, jun 2006.
- [42] Anders Wang Kristensen, Tomas Akenine-Möller, and Henrik Wann Jensen. Precomputed local radiance transfer for real-time lighting design. *ACM Trans. Graph.*, 24(3):1208–1215, 2005.
- [43] Jaroslav Krivanek, Sumanta Pattanaik, and Jiří Štěpánek. Adaptive mesh subdivision for precomputed radiance transfer. In *SCCG '04: Proceedings of the 20th spring conference on Computer graphics*, pages 106–111, New York, NY, USA, 2004. ACM Press.

-
- [44] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of ACM SIGGRAPH 97*, pages 117–126. ACM Press/Addison-Wesley Publishing Co., 1997.
- [45] Michael Lee and Hanan Samet. Navigating through triangle meshes implemented as linear quadtrees. *ACM Trans. Graph.*, 19(2):79–121, 2000.
- [46] Jaakko Lehtinen. A framework for precomputed and captured light transport. *ACM Trans. Graph.*, 26(4):13, 2007.
- [47] Jaakko Lehtinen and Jan Kautz. Matrix Radiance Transfer. In *Proceedings of the 2003 Symposium on Interactive 3D graphics*, pages 59–64, 2003.
- [48] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM Press.
- [49] Xinguo Liu, Peter-Pike Sloan, Heung-Yeung Shum, and John Snyder. All-Frequency Precomputed Radiance Transfer for Glossy Objects. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering 2004)*, pages 337–344. The Eurographics Association, 2004.
- [50] Brandon Lloyd, J. Wendt, Naga K. Govindaraju, and Dinesh Manocha. Cc shadow volumes. In *Proceedings of the 2nd EG Symposium on Rendering*, Springer Computer Science. Eurographics, Eurographics Association, 2004.
- [51] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *Proceedings of ACM SIGGRAPH 2001*, pages 519–528, August 2001.
- [52] Tobias Martin and Tiow-Seng Tan. Anti-aliasing and continuity with trapezoidal shadow maps. In *Proceedings of the 2nd EG Symposium on Rendering*, Springer Computer Science. Eurographics, Eurographics Association, 2004.
- [53] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-Frequency Shadows Using Non-linear Wavelet Lighting Approximation. *ACM Transactions on Graphics*, 22(3):376–381, 2003.
- [54] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple Product Wavelet Integrals for All-frequency Relighting. *ACM Transactions on Graphics*, 23(3):477–487, 2004.
- [55] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric Considerations and Nomenclature for Reflectance. NBS Monograph 160, National Bureau of Standards, 1977.
- [56] Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, 2004.
- [57] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [58] Ravi Ramamoorthi and Pat Hanrahan. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of ACM SIGGRAPH 2001*, pages 497–500. ACM Press, August 2001.
- [59] Walter Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, 1987.
- [60] Friedrich Sauvigny. *Partial Differential Equations 1, 1st ed.* Springer, 2006.

BIBLIOGRAPHY

- [61] Jörg Schmittler, Ingo Wald, and Philipp Slusallek. Saarcor: a hardware architecture for ray tracing. In *HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 27–36, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [62] Peter Schröder and Wim Sweldens. Spherical wavelets: efficiently representing functions on the sphere. *Computer Graphics*, 29(Annual Conference Series):161–172, 1995.
- [63] Pradeep Sen, Michael Cammarano, and Pat Hanrahan. Shadow silhouette maps. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 2003.
- [64] Peter-Pike Sloan. Normal mapping for precomputed radiance transfer. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 23–26, New York, NY, USA, 2006. ACM Press.
- [65] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered Principal Components for Precomputed Radiance Transfer. *ACM Transactions on Graphics*, 22(3):382–391, 2003.
- [66] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, 21(3):527–536, 2002.
- [67] Peter-Pike Sloan, Xinguo Liu, Heung-Yeung Shum, and John Snyder. Bi-Scale Radiance Transfer. *ACM Transactions on Graphics*, 22(3):370–375, July 2003.
- [68] Peter-Pike Sloan, Ben Luna, and John Snyder. Local, deformable precomputed radiance transfer. *ACM Trans. Graph.*, 24(3):1216–1224, 2005.
- [69] Marc Stamminger and George Drettakis. Perspective shadow maps. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2002)*, pages 557–562, 2002.
- [70] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer, part 1. *IEEE Comput. Graph. Appl.*, 15(3):76–84, 1995.
- [71] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer, part 2. *IEEE Comput. Graph. Appl.*, 15(4):75–85, 1995.
- [72] Ben Sunshine-Hill and Petros Faloutsos. Photorealistic lighting with offset radiance transfer mapping. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 15–21, New York, NY, USA, 2006. ACM Press.
- [73] Naoki Tamura, Henry Johan, and Tomoyuki Nishita. Deferred shadowing for real-time rendering of dynamic scenes under environment illumination: Image, colour and illumination in animation. *Comput. Animat. Virtual Worlds*, 16(3-4):475–486, 2005.
- [74] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.*, 25(3):967–976, 2006.
- [75] Freedon W. and Windhheuser U. Spherical wavelet transform and its discretization, 1994.
- [76] Bruce Walter, Gen Alipay, Eric Lafortune, Sebastian Fernandez, and Donald P. Greenberg. Fitting virtual lights for non-diffuse walkthroughs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 45–48, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

- [77] Liang Wan, Tien-Tsin Wong, and Chi-Sing Leung. Spherical q2-tree for sampling dynamic environment sequences. In *Proceedings of the Eurographics Symposium on Rendering Techniques, Konstanz, Germany, June 29 - July 1, 2005*, pages 21–30. Eurographics Association, 2005.
- [78] Rui Wang, John Tran, and David Luebke. All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering 2004)*, pages 345–354. The Eurographics Association, 2004.
- [79] Rui Wang, John Tran, and David Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph.*, 24(3):1202–1207, 2005.
- [80] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 265–272, New York, NY, USA, 1992. ACM Press.
- [81] Lance William. Casting curved shadows on curved surfaces. *Computer Graphics (Proceedings of SIGGRAPH '78)*, pages 270 – 274, 1978.
- [82] Michael Wimmer, D. Scherzer, and Werner Purgathofer. Light space perspective shadow maps. In *Proceedings of the 2nd EG Symposium on Rendering*, Springer Computer Science. Eurographics, Eurographics Association, 2004.
- [83] Andrew Woo, Pierre Poulin, and Alain Fournier. A survey of shadow algorithms. *IEEE Comput. Graph. Appl.*, 10(6):13–32, 1990.
- [84] Xu, Huiying, Sun, and Yinlong. Compact representation of spectral brdfs using fourier transform and spherical harmonic expansion. *Computer Graphics Forum*, 25(4):759–775, December 2006.
- [85] Hansong Zhang. Forward shadow mapping. In *Rendering Techniques '98 (Proceedings of the 9th EG Workshop on Rendering*, Springer Computer Science, pages 131–138. Eurographics, Eurographics Association, 1998.