Helsinki University of Technology
Publications in Telecommunications Software and Multimedia
Teknillisen korkeakoulun tietoliikenneohjelmistojen ja multimedian julkaisuja
Espoo 2007 TML-A17

NOVEL ILLUMINATION ALGORITHMS FOR OFF-LINE AND REAL-TIME RENDERING

Janne Kontkanen

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering, for public examination and debate in Auditorium TU1 at Helsinki University of Technology (Espoo, Finland) on the 9th of February, 2007, at 12 noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Telecommunications Software and Multimedia Laboratory

Teknillinen korkeakoulu
Tietotekniikan osasto
Tietoliikenneohjelmistojen ja multimedian laboratorio

Distribution:
Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory
P.O.Box 5400
FIN-02015 HUT
Finland
Tel. +358-9-451 2870
Fax. +358-9-451 5014
http://www.tml.tkk.fi/

Available in PDF format at http://lib.hut.fi/Diss/2007/isbn9789512286102/

ISBN 978-951-22-8609-6 (printed version)
ISSN 1456-7911
ISBN 978-951-22-8610-2 (electronic version)
ISSN 1455-9722

Otamedia Oy
Espoo 2007

ABSTRACT

Author Janne Kontkanen
Title Novel Illumination Algorithms for Off-line and Real-time Rendering

This thesis presents new and efficient illumination algorithms for *off-line* and *real-time* rendering.

The realistic rendering of arbitrary *indirect illumination* is a difficult task. Assuming *ray optics* model of light, the *rendering equation* describes the propagation of light in the scene with high accuracy. However, the computation is expensive, and thus even in off-line rendering, i.e., in pre-rendered animations, indirect illumination is often approximated as it would otherwise constitute a bottleneck in the production pipeline.

Indirect illumination can be computed using Monte Carlo integration, but when restrained to a reasonable amount of computation time, the result is often corrupted by noise. This thesis includes a method that effectively reduces the noise by applying a spatially varying filter to the noisy illumination.

For real-time performance, some components of indirect illumination can be precomputed. Irradiance volume and many variations of it pre-compute reflections and shadowing of a static scene into a volumetric data structure. This data is then used to shade dynamic objects in real-time. The practical usage of the method is limited due to aliasing artifacts. This thesis shows that with a suitable super-sampling approach, a significant quality improvement can be obtained.

Another direction is to precompute how light propagates in the scene and use the precomputed data during run-time to solve both direct and indirect illumination based on the known incident lighting. To keep the memory and precomputation costs tractable, these methods are typically restricted to infinitely distant lighting. Those that are not, require a very long precomputation time. This thesis presents an algorithm that adopts a wavelet-based hierarchical finite element method for the precomputation. A significant performance improvement over the existing techniques is obtained.

When full global illumination cannot be afforded, ambient occlusion is an attractive alternative. This thesis includes two methods for real-time rendering of ambient occlusion in dynamic scenes. The first method models the shadowing of ambient light between rigid moving bodies. The second method gives a data-oriented solution for rendering approximate ambient occlusion for animated characters in real-time. Both methods achieve unprecedented efficiency.

UDC 004.925, 004.925.3
Keywords computer graphics, shading, global illumination, indirect illumination, ambient occlusion

PREFACE

The research for this thesis was carried out during 2003-2006 in Telecommunications Software and Multimedia Laboratory of Helsinki University of Technology, Finland.

I would like to thank my thesis supervisor Lauri Savioja for both supporting my work on the thesis and our Computer Graphics Group at Helsinki University of Technology.

I am indebted to my co-author and thesis instructor Timo Aila, who is an inspiring researcher to work with and was of great help in many practical issues in finalizing the thesis. I want to thank my co-authors Nicolas Holzschuch, Samuli Laine, Alexander Keller, Jussi Räsänen, François Sillion, and Emmanuel Turquin for their significant contributions on this thesis. I want to express my gratitude to Vesa Meskanen for fruitful ideas on irradiance filtering and to Jaakko Lehtinen for always being available for discussions on mathematics or global illumination.

I want to thank my thesis pre-examiners Ulf Assarsson and Philippe Bekaert for their thorough work. Additionally, Olli Raatikainen was of great help in proofreading.

For financial support, I would like to thank Anima Vitae, Bitboys (now a subsidiary of AMD), Hybrid Graphics (now a subsidiary of NVidia Corporation), Nokia, Remedy Entertainment, and Tekes (The National Technology Agency of Finland).

Finally, I would like to thank Katariina Rautio for making my life a happy one.

Otaniemi, Espoo, 15th January 2007

Janne Kontkanen

TABLE OF CONTENTS

LIST OF PUBLICATIONS

This thesis summarizes the following articles and publications, referred to as [P1]–[P5]:

[P1] J. Kontkanen, J. Räsänen and A. Keller. Irradiance Filtering for Monte Carlo Ray Tracing. In *Monte Carlo and Quasi Monte Carlo Methods 2004*, pages 259–272. Springer-Verlag, 2004.

[P2] J. Kontkanen and S. Laine. Ambient Occlusion Fields. In *proceedings of ACM Siggraph 2005 Symposium on Interactive 3D Graphics and Games*, pages 41–48. ACM Press, 2005.

[P3] J. Kontkanen and S. Laine. Sampling Precomputed Volumetric Lighting. In *Journal of Graphics Tools*, pages 1–16, volume 11, number 3, A K Peters, 2006

[P4] J. Kontkanen, E. Turquin, N. Holzschuch and F. X. Sillion. Wavelet Radiance Transport for Interactive Indirect Lighting. In *Rendering Techniques 2006* (Eurographics Symposium on Rendering), pages 161–171. Eurographics Association, 2006.

[P5] J. Kontkanen and T. Aila. Ambient Occlusion for Animated Characters. In *Rendering Techniques 2006* (Eurographics Symposium on Rendering), pages 343–348. Eurographics Association, 2006.

LIST OF ABBREVIATIONS

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
AO	Ambient Occlusion
BRDF	Bi-directional Reflectance Distribution Function
DTO	Direct Transport Operator
FEM	Finite Element Method
GI	Global Illumination
GPU	Graphics Processing Unit
GTO	Global Transport Operator
IC	Irradiance Caching
IF	Irradiance Filtering
PRT	Precomputed Radiance Transfer

1 INTRODUCTION

1.1 Realistic Image Synthesis / Rendering

Realistic image synthesis, a branch of computer graphics, has been subject to research since the 1970s. It refers to the problem of generating an image out of a description of a 3D virtual world so that the result resembles a picture of the real world. For the illusion of realism, both the 3D model of the world and the method of computing the image need to meet certain standards.

In context of computer graphics, the word *rendering* is used interchangeably with image synthesis. We adopt this convention. *Photorealistic rendering* aims to produce photograph-like images, whereas *non-photorealistic rendering* [100] may have various different goals. As an example, non-photorealistic rendering methods may produce cartoon-like results. In this thesis, the emphasis is on realistic computation of illumination which is an essential component of photorealistic rendering.

Rendering methods are applied in a variety of different fields, such as movie production, computer games, scientific visualization, advertisements, military simulations, architecture, and medical technology. The methods developed in this thesis are primarily aimed for movie production and computer games, while they might be applicable also in other branches of the industry.

The real-world counterpart of rendering is taking a picture with a camera. The physics of this process are well understood: light is emitted from the light sources, then it propagates in the scene, and potentially ends up through the lens of the camera to the film plane making a contribution to the image. The *rendering equation* (Section 1.4) can describe this process with high accuracy[1]. A brute-force solution to the equation is possible [49], but it is often not affordable. With current PC hardware, the computation of a single high quality image may take hours or even days, depending on the scene. Thus, practical rendering methods need to deal with the conflicting goals of efficiency and accuracy.

1.2 Offline vs. Real-time Rendering

Both computer generated movies and games make use of *animation*: successively shown static images create an illusion of fluent motion. For smooth animation, the rate of images need to be high enough. It is widely accepted that 25 images per second gives a fairly good impression of motion and a lower rate may appear jerky [28].

While the problem of synthesizing images in games and movies is somewhat similar, the efficiency requirements are different. In computer generated movies, it is tolerable if one frame (i.e. an image) of animation takes minutes or even hours to render, since the computation is done *off-*

[1] Assuming the ray optics model of light. The assumptions made in this thesis are discussed in Section 1.3.

(a) (b)

Figure 1.1: Comparison between a state-of-the-art movie and computer game. **(a)** *King Kong* (Universal Studios 2005). **(b)** *King Kong*, the official game of the movie (Ubisoft Entertainment 2005).

line, prior presenting the animation to the observer. In computer games, each frame of animation is rendered on the fly, i.e., in *real-time*. This means that the synthesis should not take more than 25th of a second. In practice, this goal is reached only by compromising the quality. Despite the quick development of dedicated graphics hardware, real-time applications are still far from the quality of computer generated movies. See Figure 1.1 for a comparison between state of the art of rendering in movies and games.

The performance of interactive graphics applications is often characterized as *real-time* or *interactive*. The former refers to at least $25/s$ frame rate, whereas the latter means that the refresh may be slower than $25/s$ but fast enough so that the user can understand the series of images as movement, though the impression is not necessarily pleasant.

Both off-line and real-time methods typically apply numerous acceleration methods and approximations to meet the efficiency requirements. In practice, real-time applications often omit rendering correct soft shadows, depth of field, small-scale surface detail, and realistic surface materials. Additionally, they perform insufficient sampling in both spatial and temporal dimensions which shows up as aliasing artifacts.

A particularly long-lived approximation utilized in real-time applications and often also in the off-line domain is the lack of plausible *indirect illumination*. Indirect illumination refers to shading surfaces based on the lighting that is reflected from the other surfaces in the scene. Instead, most games and even off-line rendered animations only account for *direct illumination*, i.e., the light that arrives directly from the light sources with no intermediate surface reflections. An efficient solution for rendering indirect illumination in real-time is considered as one of the most difficult challenges of realistic image synthesis. Physically based illumination including both direct and indirect components is called *global illumination*.

Since indirect illumination is often omitted for performance reasons, other solutions have also been proposed. Traditionally, the real-time shading has been factored into direct and *ambient light*, where the latter is included to compensate for the missing indirect lighting. The ambient component is a constant for the whole scene representing the average indirect illumination. However, this makes the surfaces look somewhat dull. More recently, *ambient occlusion* techniques [121, 59, 12] have gained popularity in both real-time and off-line rendering. Ambient occlusion gives more realistic results, and is able to generate effects that have traditionally been obtained only by computing a full global illumination solution.

This thesis is composed of five publications, three of which deal with global illumination [P1,P3,P4] and two with ambient occlusion [P2,P5].

1.3 Scope and Assumptions

The new methods developed in the thesis are connected to each other in a sense that they aim for efficient computation of illumination using approximations. However, they apply rather different techniques to achieve their goals. Publication [P4] is involved with finite element methods and precomputed radiance transfer, [P3] with irradiance volume, [P1] with ray tracing based global illumination, and [P2,P5] with ambient occlusion. Thus, the related work section discusses a rather large range of rendering techniques.

On the other hand, to keep the thesis conveniently compact, basic concepts of rendering are not discussed. While global illumination, ambient occlusion, and wavelets are shortly introduced, the thesis assumes basic knowledge of computer graphics and radiometric quantities such as *radiance*, *radiosity* and *irradiance*. Also the classical ray-optics-model is adopted without further discussion and so, e.g., diffraction is not considered, light is assumed to travel at infinite speed, and we present all the equations as if they were independent of the wavelength. Furthermore, none of the methods in this thesis deal with light traveling in participating media. For background information we suggest any recent textbook, such as [79], [26], and [88]. Good sources for information on real-time rendering are [50] and [2].

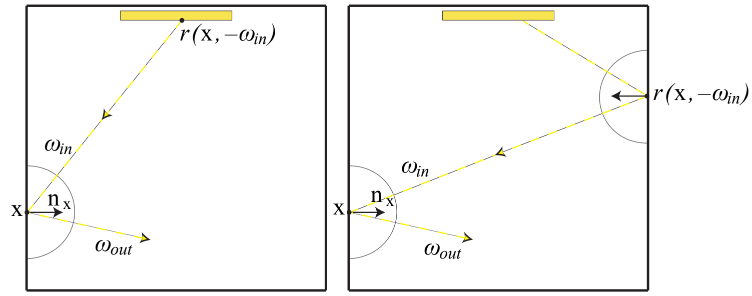
1.4 Global Illumination

Given a surface location $\mathbf{x}$ the outgoing radiance due to *direct illumination* is given by

$$L(\mathbf{x},\omega_{out}) := L_e(\mathbf{x},\omega_{out}) + \int_\Omega f_r(\omega_{in},\mathbf{x},\omega_{out}) L_e(r(\mathbf{x},-\omega_{in}),\omega_{in}) \lfloor -\omega_{in}\cdot\mathbf{n_x}\rfloor d\omega_{in}, \tag{1.1}$$

where f_r is the bi-directional reflectance distribution function capturing the material properties [26]. L_e is the radiance emitted by the surface, being non-zero only on the light sources, $r(\mathbf{x},\omega_{in})$ is a ray cast function that gives the first ray intersection if a ray is shot from location $\mathbf{x}$ towards the direction ω_{in}, and $\mathbf{n_x}$ is the surface normal at point $\mathbf{x}$. Parentheses $\lfloor$ and $\rfloor$ *clamp* the expression to zero from below, i.e., $\lfloor a\rfloor = \max(a,0)$. The above illuminates surface point $\mathbf{x}$ only by the light arriving directly from the light sources, i.e., the pre-defined set of emitting surfaces. This is illustrated in Figure 1.2a.

Direct illumination can be analytically evaluated in the case of *point light sources*, but with *area lights* numerical integration method such as Monte Carlo or Quasi Monte Carlo [54] integration is typically chosen.



(a) (b)

Figure 1.2: The difference of direct and global illumination. **(a)** Direct illumination (Equation 1.1): the radiance outgoing from the surface location $\mathbf{x}$ towards the direction ω_{out} is computed by accounting for the incident radiance that arrives directly from the light source. **(b)** Global illumination (Equation 1.2): location $r(\mathbf{x}, -\omega_{in})$ may contribute to lighting at $\mathbf{x}$ via indirect reflections even when $r(\mathbf{x}, -\omega_{in})$ does not reside on a surface of light source.

While cheaper than indirect illumination, physically correct area light sources cannot be evaluated in real-time with current PC hardware and algorithms. The cost is mainly due to *visibility*, i.e., evaluating Equation 1.1 requires finding the surface locations that are visible from the point $\mathbf{x}$. These *shadow algorithms* are out of the scope of this thesis. For more information see for example [117] and [37].

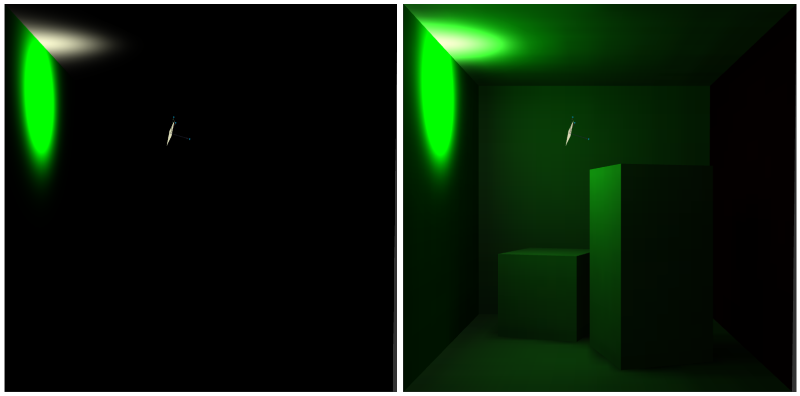
Full global illumination, on the other hand, is obtained if we also account for the reflected light, making the equation recursive:

$$L(\mathbf{x}, \omega_{out}) = L_e(\mathbf{x}, \omega_{out}) + \int_\Omega f_r(\omega_{in}, \mathbf{x}, \omega_{out}) L(r(\mathbf{x}, -\omega_{in}), \omega_{in}) \lfloor -\omega_{in} \cdot \mathbf{n_x} \rfloor d\omega_{in} \quad (1.2)$$

Notably the only change between Equations 1.1 and 1.2 is that we changed L_e to L inside the integrand. The result is called the *rendering equation* and is probably the most important equation in realistic image synthesis. It fully expresses the light propagation between the surfaces of the scene including direct and indirect illumination. This is illustrated in Figure 1.2b. The rendering equation was first introduced to computer graphics by Kajiya [49]. It can be evaluated by the Monte Carlo method using Kajiya's path tracing approach or one of the many methods discussed in Section 2.1. Sometimes accounting for indirect illumination has rather dramatic impact on the image. This is illustrated in Figures 1.3a-b.

Radiosity methods (Section 2.1) assume only diffuse reflectors/emitters, in which case Equation 1.2 can be simplified. The BRDF can now be replaced by diffuse reflectivity $f_r(\cdot, \mathbf{x}, \cdot) \rightarrow \rho(\mathbf{x})/\pi$ and radiance by radiosity $L(\mathbf{x}) \rightarrow B(\mathbf{x})/\pi$. This yields the *radiosity integral equation*:

$$B(\mathbf{x}) = B_e(\mathbf{x}) + \frac{\rho(\mathbf{x})}{\pi} \int_\Omega B(r(\mathbf{x}, -\omega_{in})) \lfloor -\omega_{in} \cdot \mathbf{n} \rfloor d\omega_{in} \quad (1.3)$$



(a) (b)

Figure 1.3: The Cornell box illuminated by a single spot light using **(a)** direct illumination and **(b)** global illumination (direct+indirect).

Where $B(\mathbf{x})$ gives the surface radiosity of point $\mathbf{x}$ and $\rho(\mathbf{x})$ is the diffuse albedo constrained into range $[0,1]$ for physically meaningful materials. A more conventional form of the above is obtained by changing to integration over surfaces:

$$B(\mathbf{x}) = B_e(\mathbf{x}) + \frac{\rho(\mathbf{x})}{\pi} \int_A B(\mathbf{y}) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) d\mathbf{y} \qquad (1.4)$$

Here $G(\mathbf{x}, \mathbf{y})$ is the *geometry term* $\frac{\lfloor \mathbf{x}\to\mathbf{y} \cdot \mathbf{n_x} \rfloor \lfloor \mathbf{y}\to\mathbf{x} \cdot \mathbf{n_y} \rfloor}{r^2_{\mathbf{xy}}}$, and $V(\mathbf{x}, \mathbf{y})$ the visibility function, being one when point $\mathbf{x}$ is visible from point $\mathbf{y}$, and zero otherwise.

1.5 Ambient Occlusion

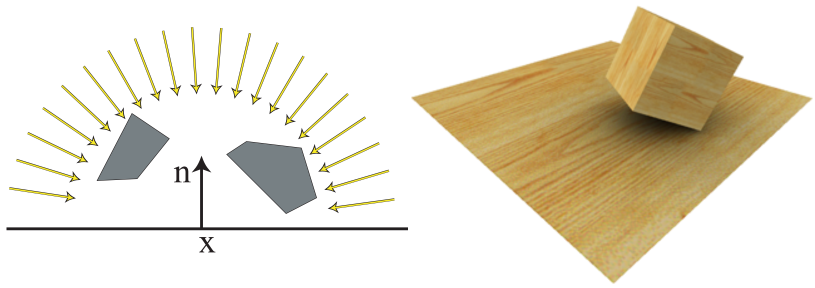
As mentioned in Section 1.2, accurate computation of global illumination is often avoided due to efficiency reasons. A classical method is to compute the direct term (in below L_{direct}) using Equation 1.1, and add the effect of constant incident ambient radiance $L_{ambient}$:

$$L(\mathbf{x}, \omega) = L_{direct}(\mathbf{x}, \omega) + \rho(\mathbf{x}) L_{ambient} \qquad (1.5)$$

Here $\rho(\mathbf{x})$ is the diffuse reflectivity and thus the above includes only the diffuse reflection of the ambient light. It would be possible to account for arbitrary reflections as well. While well known, this method is of limited use since $L_{ambient}$ does not depend on the surface geometry at all and thus gives a very flat surface appearance. For instance narrow corners receive the same amount of ambient light as open areas, while the latter should clearly be more accessible by the light.

A better approximation of indirect light is obtained using *ambient occlusion* (AO). For each surface location, AO measures how much of the hemisphere is unoccluded. Given a surface location $\mathbf{x}$, AO is defined as the hemispherical integral of visibility function:

$$A(\mathbf{x}) := \frac{1}{\pi} \int_\Omega V(\mathbf{x}, \omega) \lfloor \omega \cdot \mathbf{n_x} \rfloor \, d\omega \qquad (1.6)$$



(a) (b)

Figure 1.4: **(a)** Ambient occlusion at surface location $\mathbf{x}$ measures the portion of hemisphere accessible by the light. **(b)** Ambient occlusion gives visually convincing darkening for concave locations (the shadow cast by the cube).

Here $\mathbf{n_x}$ is the surface normal at location $\mathbf{x}$. Assuming $V = 1$ if direction ω is unoccluded and $V = 0$ otherwise, AO is proportional to the irradiance caused by constant ambient radiance arriving from all directions. Geometric illustration is shown in Figure 1.4a.

Ambient occlusion is used as a modulating coefficient for the ambient term:

$$L(\mathbf{x}, \omega) = L_{direct}(\mathbf{x}, \omega) + A(\mathbf{x})\rho(\mathbf{x})L_{ambient} \tag{1.7}$$

Accounting for AO gives a significant improvement into visual quality. Narrow corners appear darker than open areas, and objects cast realistic contact shadows on the surfaces they are resting on (see Figure 1.4b).

Computation of AO is cheaper than computing indirect illumination, but with current PC hardware it cannot be computed in real-time without approximations. Computing AO makes sense only for individual objects or for an outdoor environment, but not for indoor scenes as almost all of the hemisphere would be occluded. Thus only very little ambient light would contribute to the illumination. A possible solution to this problem is given by the *obscurances* technique [121, 68] which is discussed in Section 2.3.

Since AO depends only on the surface location and the associated normal, it can be precomputed for any rigid part of the scene and stored as vertex attributes or into a texture, and used with minimal overhead during run-time. However, if the object animates, AO needs to be re-evaluated. Different methods for computing AO are discussed in Chapter 2.3.

1.6 Wavelets

Wavelets are the building blocks of multi-resolution analysis [65]. A short introduction to wavelets is included here since wavelets are used in [P4] to construct a basis for the surface radiance and the light transport operator. However, the theory of multi-resolution analysis extends beyond the scope of this thesis – for more information about the use of wavelets in computer graphics, see for example [99] and [85].

Multi-resolution analysis using wavelets is described with two concepts: a nested set of vector spaces $V^0 \subset V^1 \subset V^2 ... \subset V^n$ spanned by *smooth*

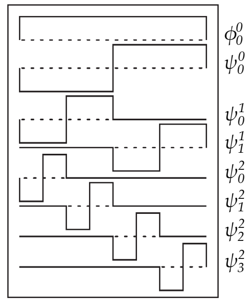


Figure 1.5: The Haar wavelet basis: single smooth function (top), and the wavelets up to the third wavelet space.

functions[2] and the associated *wavelet spaces*. If we analyze a function projected onto both V^i and V^{i+1}, the latter is capable of expressing a more detailed version of the original signal.

Given the smooth functions, the corresponding wavelet spaces and the associated wavelets are defined in terms of orthogonal complements. The wavelet space W^i is defined as orthogonal complement of V^i in V^{i+1}. Simply put, this means W^i is capable of expressing the 'part' of V^{i+1} that is missing from V^i.

The *Haar* wavelet basis is the simplest example of wavelet bases. The building blocks of the Haar basis are the following smooth function ϕ and wavelet ψ:

$$\phi(x) = \begin{cases} 1 & \text{for} \quad 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$
$$\psi(x) = \begin{cases} -1 & \text{for} \quad 0 \leq x < 1/2 \\ 1 & \text{for} \quad 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

A hierarchical set of wavelets and smooth functions of Haar basis are formed by scales and translates of the above elementary functions as follows:

$$\phi_j^i = \phi(2^i x - j), \psi_j^i = \psi(2^i x - j) \tag{1.9}$$

Where i gives the scale, and j gives the translation. Smooth functions on each level i $\phi_0^i ... \phi_n^i$ span the corresponding smooth function space V^i. Smooth functions $i+1$ form a piecewise constant basis that has twice the resolution to the scaling functions i. Since the basis functions spanning the wavelet space W^i encode the difference between the smooth function bases V^i and V^{i+1}, an arbitrary signal is fully expressed by a single smooth function ϕ_0 and the hierarchy of wavelets. The smooth function, and first three wavelet spaces are shown in Figure 1.5.

[2] often called *scaling functions*

2 RELATED WORK

2.1 Global Illumination Methods

Global illumination has been subject to research in computer graphics for decades. Dutré et al. give a complete survey of the state-of-the-art techniques [26]. Traditionally global illumination methods have been divided into two categories: finite element and ray tracing methods.

Finite Element Radiosity with Galerkin Discretization

Finite Element Method (FEM) is a general mathematical technique that discretizes a continuous domain into finite number of elements and computes an approximate solution for a given problem using the elements. For introductory purposes we will show how the classical finite element radiosity is derived from the continuous radiosity integral equation (1.4) using a technique known as *Galerkin discretization*.

First, $B(\mathbf{x})$ is approximated as a linear combination of the finite set of basis functions $\{\psi_1, \psi_2 ... \psi_n\}$:

$$B(\mathbf{x}) \approx \tilde{B}(\mathbf{x}) = \sum_i b_i \psi_i(x),$$

where b_i is the projection coefficient obtained by $b_i = \langle B|\hat{\psi}_i\rangle$. $\hat{\psi}_i$ refers to the dual basis function of ψ_i, although in case of an orthonormal basis $\hat{\psi}_i = \psi_i$. For now, we make no assumptions about the nature of the basis functions, but discussion of the possible choices follow. To obtain a method to compute coefficients b_i, we substitute $\tilde{B}(\mathbf{x})$ into Equation 1.4. This yields

$$\tilde{B}(\mathbf{x}) \approx B_e(\mathbf{x}) + \frac{\rho(\mathbf{x})}{\pi} \int_A \tilde{B}(\mathbf{y}) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) d\mathbf{y}.$$

The equation becomes an approximate equality, since the integral on the right hand side and the emitted radiosity B_e can not, in general, be exactly expressed by the basis function expansion on the left hand side. An *equation* is obtained by projecting the both sides of the approximate equality on to our set of basis functions:

$$\tilde{B}(\mathbf{x}) = \tilde{B}_e(\mathbf{x}) + \sum_i \left[\int_A \hat{\psi}_i(\mathbf{z}) \frac{\rho(\mathbf{z})}{\pi} \int_A \tilde{B}(\mathbf{y}) G(\mathbf{z}, \mathbf{y}) V(\mathbf{z}, \mathbf{y}) d\mathbf{y} d\mathbf{z} \right] \psi_i(\mathbf{x})$$

We may write the above in terms of a single basis function coefficient. We also make the assumption that the diffuse albedo $\rho(\mathbf{x})$ is constant on each element i, and can thus be replaced by ρ_i:

$$b_i = b_i^e + \frac{\rho_i}{\pi} \int_A \hat{\psi}_i(\mathbf{z}) \int_A \tilde{B}(\mathbf{y}) G(\mathbf{z}, \mathbf{y}) V(\mathbf{z}, \mathbf{y}) d\mathbf{y} d\mathbf{z}$$

Expand $\hat{B}(y)$:

$$b_i = b_i^e + \frac{\rho_i}{\pi} \int_A \hat{\psi}_i(\mathbf{z}) \int_A \left[\sum_j b_j \psi_j(\mathbf{y}) \right] G(\mathbf{z}, \mathbf{y}) V(\mathbf{z}, \mathbf{y}) d\mathbf{y} d\mathbf{z}$$

$$b_i = b_i^e + \frac{\rho_i}{\pi} \sum_j b_j \int_A \hat{\psi}_i(\mathbf{z}) \int_A \psi_j(\mathbf{y}) G(\mathbf{z}, \mathbf{y}) V(\mathbf{z}, \mathbf{y}) d\mathbf{y} d\mathbf{z}$$

In above, $\frac{1}{\pi} \int_A \hat{\psi}_i(\mathbf{z}) \int_A \psi_j(\mathbf{y}) G(\mathbf{z}, \mathbf{y}) V(\mathbf{z}, \mathbf{y}) d\mathbf{y} d\mathbf{z}$ is a *form factor* that couples the two basis functions ψ_i and ψ_j together. Denoting it with $F_{i,j}$ we arrive at the group of linear equations:

$$b_i = b_i^e + \rho_i \sum_j b_j F_{i,j} \tag{2.1}$$

Different methods to solve this and the closely related finite element formulation for arbitrary reflections are discussed in the following. However, the amount of publications that discuss applications of the FEM to global illumination is overwhelming, and only a summary is given here. For a more expanded discussion, we recommend textbooks such as [18], [90], and [26].

Finite element method was first applied to the problem of global illumination by Goral et al. [30]. The authors demonstrated solving diffuse inter-reflections in a simple environment using FEM. Soon, a plethora of new methods appeared.

The method by Goral et al. computes an approximation of continuous surface radiosity on a piecewise constant basis. Each surface patch in the scene is subdivided into the desired number of sub-elements and on each sub-element the continuous radiosity is approximated by a constant value. The continuous radiosity integral equation (1.4) is converted to the discrete form using Galerkin discretization as described. Heckbert et al. and Zatz propose higher order basis functions for more faithful re-production of radiosity with less elements [39, 119, 107].

Finding the solution for the linear system (Equation 2.1) is fairly quick using Jacobi or Gauss-Seidel iteration, for instance. Still, the traditional radiosity approach of Goral et al. is problematic since the form factors $F_{i,j}$ need to be computed and stored before the equation can be solved. The form factors can be computed by hemi-cube rasterization [17] or by ray tracing (for different methods, see [26]). With either method the computation is expensive and the storage cost is relative to the second power of the number of basis functions. As an informal estimate, accurately representing lighting even on a moderately complex scene, the number of basis functions need to be tens of thousands or even millions.

Intuitively, Jacobi or Gauss-Seidel iterations solve Equation 2.1 by successively gathering light from all the patches in the environment into each patch. The process can also be reversed: progressive radiosity algorithm computes the solution by iteratively shooting light energy to the scene [15]. This approach is related to the numerical solution method called Southwell relaxation [31].

To address the storage cost problem, many techniques try to avoid storing the full form factor matrix. Iterative solution methods often process only one row or column of the matrix at a time, in which case the form factors can be relatively efficiently computed on the fly by rendering a single hemicube, or tracing a hemispherical distribution of rays. This approach was utilized in the progressive radiosity approach by Cohen et al. [15].

Another direction is presented by stochastic radiosity methods [86, 72, 73, 8]. In these methods, the matrix-vector-products for iterative solution of the linear system are estimated by Monte Carlo integration. These methods have the advantage that the form factors do not need to be explicitly computed.

The finite element discretization of surface radiosity creates practical problems. The surface radiance is a complicated function that may be locally smooth, but then exhibit sudden discontinuities due to shadow boundaries. For accurate computation and display of radiosity, careful mesh preprocessing is needed [7]. Later research on discontinuity meshing [40, 38, 63, 98, 101] focus on identifying hard shadow boundaries and higher order discontinuities due to area lights using geometric information. Naturally, this research area is also closely related to the problem of rendering shadows [117, 37]. Worrall presented an algorithm for incremental discontinuity meshing in presence of dynamic lighting [118].

Despite the progress of discontinuity meshing based approaches, the finite element solution is often used as an intermediate step before a *per-pixel final gather* is performed to improve quality. This can be done by integrating the incident illumination at locations seen through each pixel by tracing a hemispherical distribution of rays. The precomputed finite element solution is used at the first hit points of these rays. Final gather is a costly step, but the obtained improvement in quality is typically significant. A more efficient final gathering is achieved using the information available from the finite element solution for importance sampling [9, 84].

Hierarchical Finite Elements

While the computation of form factors can be avoided using stochastic methods, at least interactions between each mutually visible element pair need to be evaluated. So without assumptions about the visibility, the theoretical worst-case performance is still quadratic to the number of patches (although Monte Carlo radiosity method reaches linear time complexity given an upper bound for the variance [87]).

This problem was addressed by hierarchical finite element methods. These methods have their roots in N-body simulations for computational physics [4]. The basic idea is that when evaluating influence of the elements to each other, any distant interaction can be evaluated between clusters of elements instead of each individual pair. The error caused by this approximation is predictable and can be easily controlled. It has been shown that given any positive error tolerance, the hierarchical FEM has linear complexity in the number of patches instead of quadratic [85].

True hierarchical FEM was first applied to the problem of radiosity computation by Hanrahan et al. [35], although a more basic two-level hierarchy had already been introduced by Cohen et al. [16] several years earlier.

The basic idea of [35] is as follows. Initially the input scene consists of big coarse surface patches (quads for easy implementation). To solve the light transport in the scene, a hierarchical representation of form factor matrix is computed. The algorithm starts by considering interaction between two coarse level elements, i.e., input patches. The elements are then subdivided if required. The algorithm measures whether the interaction between the elements is expressed well enough by a single form factor. If so, a *link* between the elements is generated, and the form factor is associated with it. Otherwise the elements are subdivided, and thus multiple new interactions need to be considered. In the end, the form factor matrix is represented by a collection of links at different levels.

The decisions whether a certain link needs to be subdivided or not, are carried out by a component called the *Oracle*. Hanrahan et al. used a simple rule based on the magnitude of the form factors: if the form factor from element A to B is larger than a specified threshold, the link needs to be refined. In addition to this F*-refinement*, the authors also present BF*-refinement*, where the criteria is based not only on the magnitude of the form factor, but also on the radiosity sent by the element. The BF-refinement is possible only with a progressive method that interleave the solution of the radiosity equation and the refinement so that an approximation of the radiosity is available during the refinement. While the F and BF-refinement methods work relatively well, they are not optimal. More efficient refinement criteria is based on the estimate of the error caused by terminating the refinement [43, 62, 96].

In hierarchical radiosity the interaction between two surface locations is always described by a single link, and thus on a single level of resolution. Due to this, a step called *push-pull* is required between iterations when a lighting solution is computed. The purpose of the push-pull is to distribute the energy received on a certain element to the parent and child elements for reflecting the light forward when the next bounce needs to be evaluated.

As discussed, hierarchical radiosity [35] refines the scene in a coarse-to-fine manner, and begins by considering each pair of coarse elements – the input description of the scene. So with respect to these coarse elements, the computational complexity of the algorithm is quadratic. For this reason, the *initial linking* cost is a problem. Many different solutions have been proposed. Holzschuch et al. show that by delaying visibility evaluation, substantial savings are possible [42].

The problem of initial linking can also be addressed by global visibility preprocessing, so that mutually invisible pairs of elements can be trivially skipped [105]. On the other hand, clustering algorithms not only speed up the visibility computations, but allow to start the refinement from coarser representation than the initial element discretization [94, 95, 116].

While the hierarchical radiosity significantly reduces the storage and computational requirements compared to the conventional radiosity FEM, the storage requirements are still quite high for big scenes. Hierarchical Monte Carlo radiosity by Bekaert et al. [8] combines the idea of hierarchical radiosity and stochastic radiosity. As in the stochastic radiosity, the form factors are taken implicitly into account by the sampling technique, and thus they do not need to be explicitly stored or computed. Another way to alleviate the problem is to re-compute form factors whenever needed as described by Stamminger et al. [97].

Wavelet Radiosity

Few years after the publication of [35], it was shown that the hierarchical radiosity is a special case of larger class of wavelet radiosity methods [32, 21]. A short introduction to wavelets is given in Section 1.6.

Theoretically the only significant difference between wavelet radiosity [32] and the original radiosity method by Goral et al. [30] is that instead of piecewise constant basis, a hierarchical basis is used. However, to fully take advantage of the hierarchical basis, wavelet radiosity utilizes coarse-to-fine refinement similar to hierarchical radiosity [35] to avoid computing data that is insignificant for the solution. In [32] Gortler et al. demonstrated the use of Haar basis, *Flatlets* and *Multi-wavelets*. Cuny et al. further study the use of higher order wavelets for radiosity [21].

By their nature, wavelets are 1D functions. However, surface radiosity is a 2D quantity and the radiosity transport operator is 4D. Two common methods to build multi-dimensional wavelet bases out of the 1D wavelets are called the *standard* and *non-standard* construction.

Non-standard construction is more common in domains such as image processing. A 2D non-standard basis consists of a single 2D smooth function, and 2D wavelets. The most decisive characteristic of non-standard basis is that supports of the smooth function and the wavelets are square-shaped. This means that the basis does not encode information for analyzing the different dimensions with different resolutions. On the other hand, 2D standard basis consists of pairwise products of all 1D basis functions. Thus also the 1D basis functions from different wavelet spaces are combined.

The hierarchical radiosity by Hanrahan et al. has been shown to correspond to Wavelet Radiosity using Haar wavelets (Section 1.6) and non-standard operator decomposition. The need for push-pull step during computation of lighting solution is actually a consequence of using non-standard construction to combine the sending and receiving bases. With the standard operator decomposition there would be no requirement for push-pull. Publication [P 4] uses the standard operator decomposition to avoid the push-pull for efficient evaluation of Neumann series of the operator.

Similarly to hierarchical radiosity, efficient application of wavelet radiosity poses requirements for the scene geometry. Roughly, wavelet radiosity can be efficiently applied only for scenes that consist of large smooth surfaces, for which a straightforward 2D parameterization exist. However, Holzschuch et al. suggest a method to alleviate this problem [41].

FEM for Arbitrary BRDFs

In computer graphics words FEM and radiosity are sometimes used interchangeably. This is misleading, since FEM can also be used to solve global illumination in case of arbitrary BRDFs. In this case the solved quantity is 4D surface light field (radiance), instead of 2D radiosity. The directional domain can be discretized with piecewise constant functions [44], spherical harmonics [89], or wavelets [13].

Publication [P4] applies hierarchical finite elements for precomputed light transport (Section 2.2), and brings these two research branches together for the first time. The precomputation stage of [P4] builds on the Wavelet Radiance algorithm [13].

Ray-Tracing

Ray tracing for computer graphics was first introduced by Appel in late 1960s [5]. Goldstein and Nagel used a similar method to simulate paths of ballistic projectiles and nuclear particles [81].

The classical form of recursive ray tracing was introduced by Kay [52] and Whitted [114]. The method rendered direct illumination, ideal mirror reflections and shadows, but it could not account for indirect illumination. Cook et al. developed distributed ray tracing [19], which uses jittered rays to simulate depth of field, motion blur, area-light sources, and indirect glossy reflections.

In 1986 Kajiya formulated the rendering equation (Section 1.4) and gave an elegant ray tracing-based algorithm capable of solving it [49]. The algorithm is known as *path tracing* and it was the first physically based ray tracing method capable of computing surface inter-reflections with arbitrary reflection distribution functions.

Ray tracing as a primitive operation is just a method to figure out the visibility between two surface points. Nowadays, there is a huge amount of efficient algorithms that aim for efficient solution of global illumination using ray tracing.

While path tracing works by generating paths that begin from the viewer, bi-directional path tracing [58, 108] generates paths from both viewer and light sources and connects them in the middle. Metropolis light transport [109], applies the Metropolis sampling algorithm [69] to estimate an image with light paths that follow the optimal probability density distribution. Photon Mapping [46, 47] traces photon paths beginning from the light sources and stores the records of photons in a Kd-tree[1] covering whole scene. Then incident radiance/irradiance at any surface location is computed by estimating the photon density in the local neighborhood.

The ray tracing based methods described above work by tracing paths from the light or from the eye. Both methods are theoretically correct and can be derived from the rendering equation or an adjoint form of it [26]. Significantly, some components of illumination are best rendered by tracing paths from the light and other components by starting from the eye. For instance *caustics* are best rendered by light paths. On the other hand, pure *light ray* tracing does not work particularly well since in most scenes, only a minority of light rays actually contribute to the image [26]. However, it is possible to guide the light paths by their potential to contribute to the image [27].

Photon mapping is typically coupled with final gathering: after the initial solution has been computed using photon tracing, it is not visualized directly – instead to estimate the incident illumination at each pixel a hemispherical distribution of rays is traced. The photon map is then used to

[1] Also known as an *axis-aligned bsp-tree*, i.e, a binary space partitioning tree that is restricted to axis-aligned splitting planes.

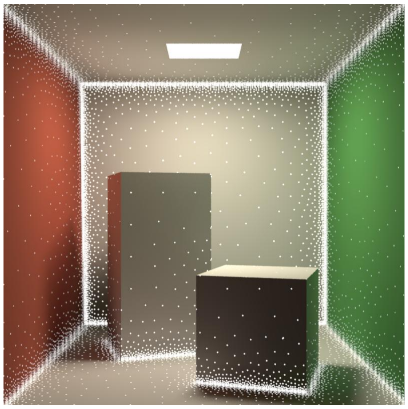


Figure 2.1: Irradiance caching computes irradiance at sparse locations (white dots), and uses interpolation to estimate the signal elsewhere. (Image courtesy of Arikan et al. [6])

compute the radiance estimates at the first hit points of these rays. Final gathering gives a significant improvement in image quality compared to the direct visualization of photon map. However, to account for the illumination due to the *caustic paths*, direct visualization is still a good idea. For the best results, two photon maps must be used: one for the caustics, and one for other indirect illumination.

Irradiance Caching and Indirect Illumination Filtering

Despite all the progress, the ray traced computation of global illumination is still a major bottleneck in off-line rendering, and real-time computation is hardly possible. Irradiance caching [113, 112] is a popular method to speed up the computation of indirect illumination: due to the tremendous speed-up it gives, it is used by most major commercial renderers. The basic idea is that the incident irradiance is computed in carefully chosen positions (only a small subset of pixels) and the remaining pixels are computed by interpolating from those sparse samples. Typical irradiance sample distribution is shown in Figure 2.1. Radiance caching [57, 56] extends the irradiance caching for general low-frequency BRDFs. For this, the full radiance distributions are computed and stored instead of irradiance.

The main strength of irradiance caching is the efficient and simple method for predicting the spatial change rate of irradiance. The predictor states that the difference of irradiance at two surface locations $\mathbf{x}_i$ and $\mathbf{x}_j$ is proportional to:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \frac{|\mathbf{x}_i - \mathbf{x}_j|}{m_i} + \sqrt{1 - \mathbf{n}_i \cdot \mathbf{n}_j} \tag{2.2}$$

In above, m_i refers to the harmonic mean distance to other visible surfaces measured on all the hemispherical directions from the location $\mathbf{x}_i$. $\mathbf{n}_i$ and $\mathbf{n}_j$ are the surface normals at the locations $\mathbf{x}_i$ and $\mathbf{x}_j$. Due to term m_i, fast changing irradiance is predicted close to corners or other concave geometry.

This is theoretically correct, but on the practical side, many authors tend to limit the density of samples to speed up the method [104, 56, 6].

While the predictor (Equation 2.2) works fairly well, it sometimes fails. As a result, the irradiance signal may be inadequately sampled and when the algorithm reconstructs the irradiance from these samples, aliasing artifacts may occur.

On the other hand there are different kinds of filtering techniques for removing noise from the irradiance signal. Jensen and Christensen conclude that median filtering is an effective way to reduce noise from the images rendered with path tracing [48]. McCool used anisotropic diffusion for the same purpose [67]. Both [48] and [67] are straightforward to implement and thus attractive in that sense. Syukens and Willems present an elegant progressive image synthesis method based on first constructing a low frequency representation of the image and successively improving the estimate with higher frequencies [102]. These methods have not become as popular as irradiance caching. Possible reasons are:

- They perform the filtering for final image (i.e. outgoing radiance), which works well only if the surface materials are smooth (counterexample: diffuse textures)
- They do not separate between indirect and direct illumination. If direct illumination filtering is needed (e.g. to filter stochastic noise due to area light sources), at least the frequency response of the filter should be different than for indirect illumination
- The methods utilize little knowledge about the scene geometry

Publication [P1] combines low pass filtering and the efficient predictor used in irradiance caching (Equation 2.2). This approach gives a smoother reconstruction of the irradiance signal and thus artifact-free images can be obtained with less computation.

After publication of [P1], an interesting alternative was found by Meyer and Anderson [70] for rendering the global illumination in *Disney-Pixar*'s animated movie *Cars*. The authors post-process animation sequences by principal component analysis (PCA). This results to a set of *eigenimages*, and eigenvalues. The key idea is that the eigenimages with the smallest eigenvalues correspond to stochastic noise, and thus by synthesizing an animation without those images, the noise is significantly reduced.

Global Illumination in Real-time

Despite the computational cost of both FEM and ray tracing there are efforts that aim to real-time rendering of global illumination.

While real-time rendering is currently dominated by rasterization-based hardware, interactive or real-time ray tracing is possible with a highly optimized algorithm [82, 111, 110]. There are efforts to perform ray tracing using custom hardware [103], and it has also been implemented using off-the-shelf rasterization hardware [80]. Recently, promising results on ray tracing with a multi-core architecture such as *Cell*[2] have been reported [10].

[2] *Cell* is a shorthand for *Cell Broadband Engine Architecture* – a microprocessor architecture jointly developed by *Sony*, *Toshiba*, and *IBM*.

Despite of the promising progress, computing a global illumination solution is still a challenge using a single PC.

The irradiance volume [34] is a different approach: the irradiance distribution function, i.e., irradiance as a function of location and surface normal is discretized into a 5D grid. This way the reflections and shadows by a static scene can be precomputed into a volume and this data can then be used during run-time to shade moving objects. A more modern form of irradiance volume is obtained by storing radiance or irradiance distribution function using spherical harmonics into each node of the 3D grid [78]. Due to the high dimensionality, low resolution needs to be used to keep the storage requirements and the precomputation time acceptable. For this reason, the method suffers from aliasing artifacts. Publication [P3] addresses this problem with a super-sampling method that divides the volume of the scene into an interesting and uninteresting domain. The method aims to faithfully produce the illumination only in the *domain of interest*, and disregards the rest of the space.

Nijasure et al. developed a real-time global illumination method [76] based on the radiance volume [34, 78]: they compute spherical harmonics representations of the incident radiance at grid nodes using GPU at run-time. Then they shade the whole scene using the radiance volume obtained this way. By performing successive iterations they are able to simulate more light bounces.

Instant radiosity [53] traces paths from the light sources and creates a number of secondary lights to represent the indirect lighting. Then the contributions of these secondary light sources are accumulated together using graphics hardware with shadow mapping [115] or shadow volumes [20]. Reflective shadow mapping [22, 23] builds on this idea, but replaces the light path tracing by rendering a view from each light source using GPU. Then, the resulting 'pixel lights' are used to illuminate the image to get a one bounce indirect illumination. Real-time performance is reached, but since only part of the pixel lights are allowed to contribute on each location in the image, the method is biased.

Radiance cache splatting [29] is a radiance caching algorithm that splats the contribution of each radiance sample into the image instead of iterating through the pixels and querying close-by samples from a spatial data structure as is conventionally done in irradiance caching algorithm. Consequently, the algorithm is more suitable for an efficient GPU implementation.

2.2 Precomputed Radiance Transfer

Lately, precomputed radiance transfer (PRT) methods have gained a great amount of attention. They are based on the idea that instead of computing everything during run-time, it is possible to precompute the light transport from an emission basis to the surfaces of the scene.

While methods that fall into the PRT-framework [60] had been developed earlier [1, 24, 25, 77, 106], Sloan et al. published an influential publication [92] in 2002. The authors precomputed transport from an infinitely distant environment lighting to the surfaces of the object. Both the emission and the outgoing radiance at each surface location were expressed in the spherical harmonics basis. The research has inspired a lot of subsequent work [74, 75, 93, 91, 64, 61, 33]. An interested reader can find more information about the PRT methods from the master's thesis of Lehtinen [60].

Following Sloan et al. [92], most of the PRT methods have the restriction that the lighting needs to be infinitely distant. However, as noted by Lehtinen [60] this is not really necessary, and in fact there already exists some work that does not adopt this restriction as explained in the following. First step towards this direction was taken by Annen et al. [3]. The authors approximate spatially varying incident lighting using first order Taylor series expansion and interpolation between multiple sampling locations.

Kristensen et al. precomputed light transport from a volumetric unstructured point cloud to the surfaces of the scene [55]. During run-time they project their dynamic light sources to the point cloud basis, apply the precomputed operator to get the indirect lighting on the surfaces. As a result they are able to render real-time global illumination with local lighting i.e. they are not restricted to infinitely distant lighting.

During the precomputation, they first compute the light transport operator with high accuracy and then, as a post-process, perform lossy compression. This means that a significant amount of data is first computed and then thrown away. Due to this, the precomputation stage of the method by Kristensen et al. is rather costly. Publication [P4] presents a method that computes local light transport without heavy precomputation by avoiding the computation of insignificant data in the first place.

Hasan et al. present a method for precomputing direct-to-indirect transport [36]. This work is both closely related and concurrent to [P4]. They precompute direct-to-indirect transport operator from the points distributed on the surfaces to the screen pixels, achieving high quality real-time global illumination, but with fixed a viewpoint. Similarly to [P4], the method also avoids the computation of insignificant interactions with a hierarchical method. [P4], however, is not restricted to fixed viewpoint.

Also publication [P2] has a connection to PRT methods. In [92], Sloan et al. present *neighborhood transfer* as an application of their PRT method. The authors precompute the transfer from the infinitely distant emission to the neighborhood of the object, i.e. to the nearby space. In neighborhood transfer, the PRT coefficients are stored into a regular 3D grid, and the authors demonstrate rendering inter-reflections and shadows using their method. [P2] can be seen as an efficient method to compute neighborhood transfer for ambient light.

2.3 Ambient Occlusion Methods

Accessibility shading [71] can be seen as a predecessor to ambient occlusion. While the definition of *accessibility* is slightly different, conceptually the measure is fairly close to ambient occlusion. Accessibility is typically used to model things such as gathering of dirt, polishing or weathering. Ambient occlusion can be understood as computing the accessibility of light.

The name *ambient occlusion* was popularized when Industrial

Light&Magic and Pixar published their new shading techniques in SIGGRAPH 2002 [59, 12], but a closely related *obscurances*-method had been published a few years earlier by Zhukov et al. [121]. The obscurance, W, of a surface location is defined as

$$W(\mathbf{x}, \mathbf{n}) := \frac{1}{\pi} \int_{\Omega} \rho(d(\mathbf{x}, \omega)) \lfloor \omega \cdot \mathbf{n} \rfloor \, d\omega, \tag{2.3}$$

where $d(\mathbf{x}, \omega)$ refers to the distance of the first intersection when a ray is shot from $\mathbf{x}$ towards ω and ρ is a function that weights the distance suitably. Iones et al. [45] suggest $1 - e^{\tau d}$, where τ is a user defined constant. Exact definition of ambient occlusion is obtained if ρ is replaced with a visibility function (Section 1.5). Mendez et al. extend obscurances to account for color bleeding by taking the reflectances of the obscuring surfaces into account [68].

To compute ambient occlusion, Equation 1.6 needs to be evaluated. This can be done using ray tracing to shoot a hemispherical distribution of rays from each surface location. Another option would be to rasterize hemi-cubes using graphics hardware. However, these kinds of brute-force methods are not suitable for real-time or interactive use.

There are several methods for speeding up rendering ambient occlusion without precomputation. Sattler et al. demonstrate quick computation of AO using graphics hardware [83]. They also utilize coherence in the visibility function to achieve interactive frame rates with deformable objects in dynamic distant illumination. Kautz et al. developed an efficient hemispherical rasterizer based on a look-up-table [51]. They also use a two-level hierarchy of triangle data to further speed up the computation. As a result, they compute shadows from low-frequency lighting environments at interactive rates. Bunnell computes approximate AO by modeling the shadow casting surface as disks [11] and evaluates the AO caused by the disks using an analytic method. He uses a heuristic method to combine the shadows cast from multiple disks. While these methods are fairly efficient, with current hardware they still require too much computational resources for demanding real-time applications.

Publication [P2] presents a real-time technique that can be used for rendering AO shadows cast by rigid bodies to other surfaces in the neighborhood. The method precomputes a field around each rigid shadow caster, and uses this data to render the shadows in real-time. Zhou et al. presented a concurrent work [120] that is closely related to [P2]. Instead of just ambient shadows, the method is aimed at evaluating general precomputed shadows. The visibility as a function of direction is stored in multiple points around the shadow caster. The visibility function is represented by Haar wavelets or spherical harmonics. The contribution of multiple objects can be handled more accurately than in [P2], but with higher cost. Malmer et al. [66] further developed [P2]: they use regular grid instead of the cube-map used in [P2], and solve several practical issues such as how to automatically choose the resolution of the grid.

[P2] has the restriction that while the shadow casting objects can move relative to receiving surfaces, the casters must be rigid, i.e., they cannot deform in any way. Publication [P5], on the other hand, presents a real-time method for approximating ambient occlusion on an animated character.

3 NEW TECHNIQUES FOR GLOBAL ILLUMINATION

This chapter describes new techniques for efficient rendering of global illumination. Publication [P1] removes noise from indirect irradiance computed by Monte Carlo ray tracing, allowing to compute the irradiance with fairly few integration samples. [P3] describes a method that reduces aliasing of precomputed volumetric lighting. [P4] uses a hierarchical finite element method to efficiently precompute radiance transport in a scene. Publications about ambient occlusion, i.e., [P2] and [P5], are discussed in Chapter 4.

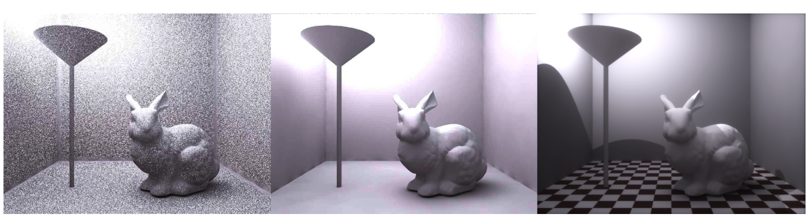
3.1 Irradiance Filtering [P1]

It is possible to generate photorealistic images with global illumination using stochastic ray tracing methods, such as path tracing [49], bi-directional path tracing [58, 108], or photon mapping [46]. However, a rather long computation time is required to keep the noise on a visually acceptable level. Most of the noise is due to insufficient amount of Monte Carlo samples used to compute the irradiance due to indirect reflections.

Irradiance filtering (IF) builds on the assumption that most of the energy of the indirect irradiance signal lies in spatially low frequencies, and that applying a suitable low pass filter on this signal would improve the signal-to-noise ratio and the visual appearance of the image. Since the frequency content of irradiance is spatially varying, the filter needs to adapt to local characteristics of the scene. To predict the local frequency content of the signal, IF uses the successful predictor developed by Ward et al. [113] for irradiance caching (IC) (Section 2.1, Equation 2.2).

The difference to IC is that instead of computing accurate irradiance values at sparse locations (Figure 2.1), IF computes noisy values at dense locations, i.e., at every pixel. The noisy irradiance signal is shown in Figure 3.1a.

On the locations where the irradiance signal is expected to change fast according to the predictor, IF computes the irradiance estimates more



(a) (b) (c)

Figure 3.1: The irradiance filtering algorithm. **(a)** Noisy indirect irradiance computed by Monte Carlo ray tracing. **(b)** Filtered irradiance. **(c)** Outgoing radiance obtained by multiplying with diffuse albedo and combining with the direct illumination.

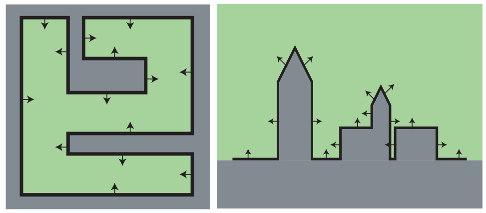


Figure 3.2: Domain of interest refers to the volume from which back-faces of the scene geometry are not visible (green).

accurately by using more integration rays and only a small filter can be used in these neighborhoods to preserve the detail in the signal. See, for example, the corners in Figure 3.1a. On the other hand, on flat open areas irradiance estimates can be computed with very few integration rays since a large filter kernel is used for noise removal. For the resulting quality, see Figures 3.1b-c.

IF algorithm is biased, i.e., the expectation value of the result is slightly incorrect. This is because the noisy irradiance signal is sampled at pixel locations, which results into uneven distribution of samples in the world space. Additionally, the predictor by Ward et al. sometimes fails to correctly predict the frequency content. Despite of this, IF typically gives smooth, visually pleasing re-construction in scenes where IC generates discontinuous artifacts with the same amount of computation.

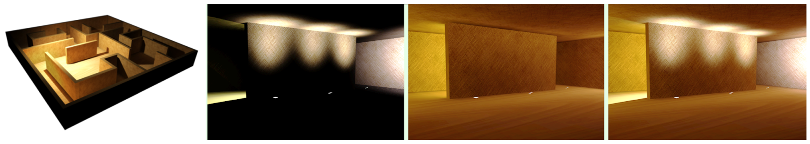
3.2 Sampling Precomputed Volumetric Lighting [P3]

In the irradiance volume [34], neighborhood transfer [92], and several shadowing techniques [66, 120][P2], lighting information is precomputed into a volume to quickly shade moving objects. Often, a regular grid data structure is used for storage due to its simplicity. However, low spatial resolution needs to be used to keep the memory requirements and precomputation cost acceptable. For this reason, precomputed volumetric lighting often suffers from aliasing artifacts.

In irradiance volume, an irradiance distribution function is precomputed and stored into the nodes of a volumetric grid. Each node of the grid thus stores the irradiance as a function of normal of (imaginary) receiving surface. The precomputation can be done by rasterizing cube-maps as seen from the locations of the nodes or using ray tracing. We will refer to this method as *point sampling* to reflect that the precomputation is done at the exact points defined by the nodes of the grid.

Theoretically, to achieve signal reconstruction without aliasing artifacts the signal would need to be point-sampled with twice the rate of the highest frequency present (Nyquist-Shannon sampling theorem). However, lighting is not band-limited at all since the scene geometry slicing through the volume causes discontinuities. In other words, there is no sampling rate high enough to capture all the detail.

Publication [P3] proposes a simple and practical method that effectively



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

Figure 3.3: Rendering interactive/real-time global illumination using wavelet radiance transport: (**a**) overview image of the scene, (**b**) direct illumination rendered using shadow mapping with GPU, (**c**) indirect illumination obtained by applying precomputed operator to direct illumination, and (**d**) direct+indirect illumination combined.

diminishes the aliasing problem. The method is based on two ideas. First, the illumination is *super-sampled* and averaged over the tri-linear weighting functions associated with each node of the grid. This process is similar to the classic stochastic super-sampling used to reduce the pixel-level aliasing in ray tracing and rasterization. Second, we restrict the computation to the *domain of interest*. This means that we are not interested in all the volume. The motivation is that in many applications the polygonal geometry represents the boundary between open space and solid structures, and the illumination inside the solid structures is not needed. This removes most of the discontinuities caused by scene geometry slicing through the volume.

We define the *domain of interest* as every location in space from which the back-faces of the polygons are not visible. For schematic illustration of this see Figure 3.2. This definition has the advantage that given a scene geometry it is implicit, and the geometry does not have to be closed or watertight. The latter would be a frustrating requirement for a practical application. As shown in publication [P3], the new sampling method reduces aliasing significantly with only a small overhead in the precomputation time. Additionally, the run-time component does not have to be changed at all.

3.3 Wavelet Radiance Transport for Interactive Indirect Lighting [P4]

Precomputed radiance transport methods (Section 2.2) precompute the operator that expresses the relationship between light emitted from the light sources, and light outgoing from the surfaces. The precomputed operator captures shadowing and inter-reflections, and thus real-time global illumination can be achieved using these methods. As discussed (Section 2.2), many PRT methods restrict themselves to infinitely distant lighting. Currently those methods that do not have this limitation, either require a long precomputation time [55], or support only a fixed view-point [36].

Publication [P4] uses a hierarchical FEM to precompute a surface-to-surface transport operator quickly in a hierarchical wavelet basis. Then the precomputed operator is used during run-time to convert direct illumination into indirect. The method is summarized in Figures 3.3a-d. The precomputation first computes *direct transport operator* (DTO) using a method that closely resembles the wavelet radiance algorithm [14]. DTO expresses a single bounce of light in the scene. A *global transport operator* (GTO) describing converged radiance transport in the scene is then computed using Neumann series of the DTO. The run-time stage projects the emission from the dynamic light sources to the surfaces, applies the precomputed GTO to get the converged indirect lighting, and combines the result with direct light rendered by traditional techniques such as shadow mapping.

Since the method avoids computing insignificant interactions using hierarchical precomputation, a significant speed-up compared to a previous method [55] is achieved. However, due to hierarchical finite elements, the new method also suffers from restrictions: the elements need to be defined on the surfaces of the scene. Due to this [P4] uses simple scenes that initially consist of large quads.

4 NEW TECHNIQUES FOR AMBIENT OCCLUSION

This chapter discusses two new methods for rendering ambient occlusion (AO). Both are based on precomputation and support dynamic, but a restricted set of scenes.

Publication [P2] presents a method for rendering ambient shadows cast by a rigid object onto the surrounding geometry (Section 4.1), whereas [P5] is a method for rendering approximate AO for animated characters (Section 4.2).

4.1 Ambient Occlusion Fields [P2]

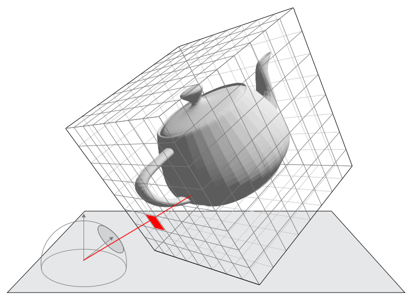


Figure 4.1: The solid angle Ω subtended by an object and the average direction of occlusion Υ are stored for each direction as functions of distance r. At run-time these functions are fetched from a cube-map and evaluated at the receiving surface in order to compute the ambient occlusion.

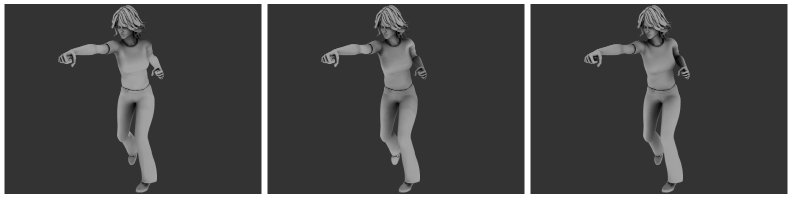
In ambient occlusion fields, the AO cast by a rigid object is precomputed as a field around the object. Then this field is used during run-time to render ambient shadows into the neighborhood.

The method builds on an idea that to compute a fairly good approximation of AO it suffices to know an approximate direction of the occluding object (Υ), and the solid angle (Ω) it occupies as seen from a shaded point. In the precomputation stage, these two pieces of information are computed and stored into the surrounding space of the object.

During run-time, AO can be computed at any receiving location as follows. First, Υ and Ω are read from the precomputed data. Then AO caused by the spherical cap with a central direction Υ and solid angle Ω is read from a small look-up table. In other words, the shadow casting object is approximated by a spherical cap at each receiving location. To further diminish the storage requirements, functions Υ and Ω are not stored into a regular grid but as simple radial functions into a cube-map.

An illustration of the technique is shown in Figures 4.1 and 1.4b. The method renders convincing AO shadows with little memory requirements and the run-time cost should be small enough for real-time applications such as computer games.

4.2 Ambient Occlusion for Animated Characters [P5]



(a) (b) (c)

Figure 4.2: Ambient occlusion computed using different methods. **(a)** AO as an average over the allowed pose space (static values that do not depend on the pose). **(b)** AO using the method explained in [P5]. **(c)** Reference solution obtained by ray tracing 1000 rays from each vertex to all hemispherical directions. The difference is best visible on the left arm of the character.

Publication [P5] is a method for approximating AO for animated characters. Characters are typically animated using parameters such as joint angles. The possible domain of poses spanned by the character is called the *pose space*.

In practice, a static ambient occlusion solution computed for either a default pose, or as an average over the pose space is often almost sufficient for a range of poses in an animation. However, the illusion of correct AO is shattered in some places, see for example the arms in the Figure 4.2a. However, actually re-computing the AO for each pose during run-time seems unnecessary.

Publication [P5] improves the approximation with little overhead on the run-time computation. This is done by establishing a mapping between the animation parameters and the AO. During precomputation stage a number of random reference poses (for example 1000) are generated and the AO is computed for all of these poses. Then an overdetermined system of linear equations is solved to find out the linear dependence between the animation parameters and the AO.

With the linear model, the method does not match the reference solution. In particular, it is unable to render effects that have non-linear dependence on the animation parameters, and effects that depend on multiple animation parameters. So, for instance, the shadows cast by the wrists of the character to the torso are not captured. Despite of this, the new method gives an approximation that both looks more plausible and is numerically closer to the reference than the static solution. Further, the cost of applying the new technique should be low enough for demanding real-time applications like computer games. A character rendered with the new method is shown in Figure 4.2b and the corresponding reference in Figure 4.2c.

5 MAIN RESULTS AND CONTRIBUTIONS OF THE AUTHOR

[P1] Irradiance Filtering for Monte Carlo Ray Tracing

Irradiance Filtering is proposed as an alternative for the widely used irradiance caching technique [113, 112]. Instead of computing costly irradiance samples in sparse locations and interpolating as in irradiance caching, the new technique computes coarse estimates in every pixel and uses a spatially varying filter to remove the noise. The initial motivation for the idea was given by Vesa Meskanen and the further development was done by the author and Jussi Räsänen. The author made half of the implementation and wrote half of the article. Alexander Keller provided support and participated in writing the final version of the paper.

[P2] Ambient Occlusion Fields

A new method for rendering ambient shadows cast by rigid objects is presented. The method precomputes a field around each object, which is then used to approximate shadows cast by the object on nearby surfaces. The method was developed by the author and Samuli Laine in equal proportions. The author wrote the article, while Mr. Laine implemented the technique.

[P3] Sampling Precomputed Volumetric Lighting

A new method to precompute lighting into a volumetric grid is presented. The method avoids aliasing artifacts even when a low resolution grid is used, while adding only a small overhead on the precomputation time. This method can be used to improve irradiance volume [34], neighborhood transfer [92], and ambient occlusion techniques [66]. The author developed the idea of super-sampling and domain of interest. Samuli Laine contributed the point classification algorithm. The author implemented the method and wrote the article.

[P4] Wavelet Radiance Transport for Interactive Indirect Lighting

A hierarchical method for precomputed light transport is presented. This work combines techniques from two separate research branches: hierarchical finite element techniques for global illumination and precomputed light transport. Most of the research was done together with Emmanuel Turquin. The hybrid 4D wavelet basis was proposed by Mr. Turquin, while the author contributed the hierarchical precomputation algorithm. The author made the implementation, while the article was written together with Mr. Turquin, Nicolas Holzschuch, and François Sillion.

[P5] Ambient Occlusion for Animated Characters

A novel method for rendering ambient occlusion for animated characters is presented. The method precomputes AO for a number of random reference poses and then establishes a linear relationship between the animation parameters and an approximate AO value at each vertex. This mapping is used during run-time to synthesize approximate AO for any pose. The author developed the method and implemented it, while Timo Aila provided insight and wrote most of the article.

BIBLIOGRAPHY

[1] John M. Airey, John H. Rohlf, and Jr. Frederick P. Brooks. Towards image realism with interactive update rates in complex virtual building environments. In *Proceedings of ACM SIGGRAPH 1990 Symposium on Interactive 3D Graphics*, pages 41–50. ACM Press, 1990.

[2] Tomas Akenine-Möller and Eric Heines. *Real-Time Rendering, 2nd Edition*. AK Peters, 2002.

[3] Tomas Annen, Jan Kautz, Frédo Durand, and Hans-Peter Seidel. Spherical Harmonic Gradients for Mid-Range Illumination . In *Rendering Techniques 2004 (Proceedings of Eurographics Symposium on Rendering)*, pages 331–336, 2004.

[4] Andrew W. Appel. An efficient program for many-body simulation. *SIAM Journal on Scientific and Statistical Computing*, 6(1):85–103, 1985.

[5] Arthur Appel. Some techniques for shading machine renderings of solids. In *AFIPS 1968 Spring Joint Computer Conf.*, volume 32, pages 37–45, 1968.

[6] Okan Arikan, David A. Forsyth, and James F. O'Brien. Fast and detailed approximate global illumination by irradiance decomposition. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, 24(3), 2005.

[7] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, pages 51–60, 1991.

[8] Philippe Bekaert. *Hierarchical and Stochastic Algorithms for Radiosity*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, 1999.

[9] Philippe Bekaert, Philip Dutré, and Yves D. Willems. Final radiosity gather step using a monte carlo technique with optimal importance sampling. Technical Report CW275, Katholieke Universiteit Leuven, 1998.

[10] Carsten Benthin, Ingo Wald, Michael Scherbaum, and Heiko Friedrich. Ray Tracing on the CELL Processor. In *Proceedings of the 2006 IEEE Symposium on Interactive Ray Tracing*, 2006. (accepted for publication, minor revision pending).

[11] Michael Bunnell. Dynamic ambient occlusion and indirect lighting. In *GPU Gems 2*, pages 223–234. Addison Wesley, 2005.

[12] Per H. Christensen. Note #35: Ambient Occlusion, Image-Based Illumination, and Global Illumination. *PhotoRealistic RenderMan Application Notes*, 2002.

[13] Per H. Christensen, Eric J. Stollnitz, and David H. Salesin. Global illumination of glossy environments using wavelets and importance. *ACM Transactions on Graphics*, 15(1):37–71, 1996.

[14] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Wavelet Radiance. In *Proceedings of Eurographics Workshop on Rendering 94*, pages 287–302, 1994.

[15] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics (Proceedings of ACM SIGGRAPH 88)*, pages 75–84, 1988.

[16] Michael F. Cohen, David S. Immel Donald P. Greenberg, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, pages 26–35, 1986.

[17] Michael F. Cohen and Donald P. Greenberg. The hemi-cube: a radiosity solution for complex environments. *Computer Graphics (Proceedings of ACM SIGGRAPH 85)*, pages 31–40, 1985.

[18] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Morgan Kaufmann Publishers, 1993.

[19] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, 18:137–145, 1984.

[20] Franklin C. Crow. Shadow algorithms for computer graphics. *Computer Graphics (Proceedings of ACM SIGGRAPH 77)*, 11(3), 1977.

[21] François Cuny, Laurent Alonso, and Nicolas Holzschuch. A novel approach makes higher order wavelets really efficient for radiosity. *Computer Graphics Forum (Proceedings of Eurographics 2000)*, 19(3):99–108, 2000.

[22] Carsten Dachsbacher and Marc Stamminger. Reflective shadow maps. In *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games*, pages 203–231. ACM Press, 2005.

[23] Carsten Dachsbacher and Marc Stamminger. Splatting indirect illumination. In *Proceedings of ACM SIGGRAPH 2006 Symposium on Interactive 3D Graphics and Games*, pages 93–100. ACM Press, 2006.

[24] Yoshinori Dobashi, Kazufumi Kaneda, Hideki Nakatani, and Hideo Yamashita. A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum (Proceedings of Eurographics 95)*, 14(3):229–240, 1995.

[25] Julie Dorsey, François X. Sillion, and Donald P. Greenberg. Design and simulation of opera lighting and projection effects. *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, pages 41–50, 1991.

[26] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. A K Peters Ltd, 2003.

[27] Philip Dutré and Yves D. Willems. Importance-driven monte carlo light tracing. In *Proceedings of Eurographics Workshop on Rendering 94*, pages 287–302, 1994.

[28] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice, Second Edition*, chapter 21, pages 1057–1058. Addison-Wesley, 1990.

[29] Pascal Gautron, Jaroslav Kivánek, Kadi Bouatouch, and Sumanta Pattanaik. Radiance cache splatting: GPU-friendly global illumination algorithm. In *Rendering Techniques 2005 (Proceedings of Eurographics Symposium on Rendering)*, 2005.

[30] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, 18:213–222, 1984.

[31] Steven Gortler, Michael F. Cohen, and Philipp Slusallek. Radiosity and relaxation methods. *IEEE Computer Graphics and Applications*, 14(6):48–58, 1994.

[32] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet Radiosity. In *Proceedings of ACM SIGGRAPH 93*, pages 221–230. ACM Press, 1993.

[33] Paul Green, Jan Kautz, Wojciech Matusik, and Frédo Durand. View-dependent precomputed light transport using nonlinear gaussian function approximations. In *Proceedings of ACM SIGGRAPH 2006 Symposium on Interactive 3D Graphics and Games*, pages 7–14. ACM Press, 2006.

[34] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. The irradiance volume. *IEEE Computer Graphics and Applications*, 18(2):32–43, 1998.

[35] Pat Hanrahan, David Salzman, and Larry Aupperle. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, 25(4):197–206, 1991.

[36] Milos Hasan, Fabio Pellacini, and Kavita Bala. Direct-to-indirect transfer for cinematic relighting. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006)*, 25(3), 2006.

[37] Jean-Marc Hasenfratz, Marc Lapierre, Nicolas Holzschuch, and François X. Sillion. A Survey of Real-Time Soft Shadows Algorithms. *Computer Graphics Forum*, 22(4), 2003.

[38] Paul Heckbert. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, University of Bristol, 1991.

[39] Paul S. Heckbert and James M. Winget. Finite element methods for global illumination. Technical report, University of California at Berkeley, 1991.

[40] David Hedley. *Discontinuity Meshing for Complex Environments*. PhD thesis, University of Bristol, 1998.

[41] Nicolas Holzschuch, François Cuny, and Laurent Alonso. Wavelet radiosity on arbitrary planar surfaces. In *Rendering Techniques 2000 (Proceedings of Eurographics Workshop on Rendering)*, pages 161–172, 2000.

[42] Nicolas Holzschuch, François X. Sillion, and George Drettakis. An Efficient Progressive Refinement Strategy for Hierarchical Radiosity. In *Proceedings of Eurographics Workshop on Rendering 94*, pages 343–357, 1994.

[43] Alvin Theophilus Campbell III. *Modeling global diffuse illumination for image synthesis*. PhD thesis, University of Texas at Austin, 1992.

[44] Dave S. Immel, Michael Cohen, and Donald P. Greenberg. A Radiosity Method for Non-Diffuse Environments. *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, 20(4):133–142, 1986.

[45] Andrey Iones, Anton Krupkin, Mateu Sbert, and Sergey Zhukov. Fast, Realistic Lighting for Video Games. *IEEE Computer Graphics and Applications*, 23(3):54–64, 2003.

[46] Henrik W. Jensen. Global Illumination Using Photon Maps. In *Rendering Techniques 96 (Proceedings of Eurographics Workshop on Rendering)*, pages 21–30, 1996.

[47] Henrik W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. AK Peteres, 2001.

[48] Henrik W. Jensen and Niels J. Christensen. Optimizing path tracing using noise reduction filters. In *Proceedings of WSCG 1995*, 1995.

[49] James T. Kajiya. The rendering equation. *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, 20, 1986.

[50] Jan Kautz. Hardware Lighting and Shading: a Survey. *Computer Graphics Forum*, 23(1):85–112, 2004.

[51] Jan Kautz, Jaakko Lehtinen, and Timo Aila. Hemispherical Rasterization for Self-Shadowing of Dynamic Objects. In *Rendering Techniques 2004 (Proceedings of Eurographics Symposium on Rendering)*, pages 179–184, 2004.

[52] Douglas S. Kay. Transparency, refraction and ray tracing for computer synthesized images. Master's thesis, Cornell University, 1979.

[53] Alexander Keller. Instant radiosity. *Computer Graphics (Proceedings of ACM SIGGRAPH 97)*, pages 49–56, 1997.

[54] Alexander Keller. *Quasi-Monte Carlo Methods for Photorealistic Image Synthesis*. PhD thesis, University of Kaiserslautern, 1998.

[55] Anders Wang Kristensen, Tomas Akenine-Möller, and Henrik Wann Jensen. Precomputed local radiance transfer for real-time lighting design. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, 24(3), 2005.

[56] Jaroslav Krivanek, Kadi Bouatouch, Sumanta Pattanaik, and Jiri Zara. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Rendering Techniques 2006 (Proceedings of Eurographics Symposium on Rendering)*, 2006.

[57] Jaroslav Krivanek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics (Accepted for publication)*, 2005. Also available as Technical Report #1623, IRISA, http://graphics.cs.ucf.edu/RCache/index.php.

[58] Eric P. Lafortune and Yves D. Willems. Bi-directional Path Tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, 1993.

[59] Hayden Landis. RenderMan in Production, ACM SIGGRAPH 2002 Course 16, 2002.

[60] Jaakko Lehtinen. Foundations of precomputed radiance transfer. Master's thesis, Helsinki University of Technology, 2004.

[61] Jaakko Lehtinen and Jan Kautz. Matrix radiance transfer. In *Proceedings of ACM SIGGRAPH 2003 Symposium on Interactive 3D Graphics*, pages 59–64. ACM Press, 2003.

[62] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. *Computer Graphics (Proceedings of ACM SIGGRAPH 94)*, pages 67–74, 1994.

[63] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Discontinuity Meshing for Accurate Radiosity. *IEEE Computer Graphics and Applications*, 12(6):25–39, 1992.

[64] 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 Eurographics Symposium on Rendering)*, 2004.

[65] Stéphane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[66] Mattias Malmer, Fredrik Malmer, Ulf Assarsson, and Nicolas Holzschuch. Fast precomputed ambient occlusion for proximity shadows. Technical Report RR-5779, INRIA, 2005.

[67] Michael D. McCool. Anisotropic diffusion for monte carlo noise reduction. *ACM Transactions on Graphics (TOG)*, 18(2):171–194, 1999.

[68] Àlex Méndez, Mateu Sbert, and Jordi Catà. Real-time Obscurances with Color Bleeding. In *Proceedings of the 19th spring conference on computer graphics*, pages 171–176, 2003.

[69] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, June 1953.

[70] Mark Meyer and John Anderson. Statistical acceleration for animated global illumination. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006)*, 2006.

[71] Gavin Miller. Efficient algorithms for local and global accessibility shading. In *Proceedings of ACM SIGGRAPH 94*, pages 319–326, 1994.

[72] Laszlo Neumann, Martin Feda, Manfred Kopp, and Werner Purgathofer. A new stochastic radiosity method for highly complex scenes. In *Proceedings of Eurographics Workshop on Rendering 94*, pages 195–206, 1994.

[73] Laszlo Neumann, Werner Purgathofer, Robert F. Tobler, Attila Neumann, Pavol Elias, Martin Feda, and Xavier Pueyo. The stochastic ray method for radiosity. Technical Report TR-186-2-95-3, Vienna University of Technology, 1995.

[74] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, 2003.

[75] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004)*, 2004.

[76] Mangesh Nijasure, Sumanta N. Pattanaik, and Vineet Goel. Real-Time Global Illumination on GPUs. *Journal of Graphics Tools*, 10(2):55–71, 2005.

[77] Jeffry S. Nimeroff, Eero Simoncelli, and Julie Dorsey. Efficient Re-rendering of Naturally Illuminated Environments. In *Rendering Techinques 94 (Proceedings of Eurographics Workshop on Rendering)*, pages 359–373. Springer-Verlag, 1994.

[78] Chris Oat. Irradiance Volumes for Games, Presentation at Game Developers Conference, 2005.

[79] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation.* Morgan Kaufmann Publishers Inc., 2004.

[80] Timothy John Purcell. *Ray tracing on a stream processor.* PhD thesis, Stanford University, 2004.

[81] Goldstein R.A. and Nagel R. 3-D Visual Simulation. *Simulation*, pages 25–31, 1 1971.

[82] Alexander Reshetov, Alexei Soupikov, and Jim Hurley. Multi-level ray tracing algorithm. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, 2005.

[83] Mirko Sattler, Ralf Sarlette, Gabriel Zachmann, and Reinhard Klein. Hardware-accelerated ambient occlusion computation. In *Proceedings of VMV '04*, pages 119–135, 2004.

[84] Annette Scheel, Marc Stamminger, and Hans-Peter Seidel. Thrifty final gather radiosity. In *Rendering Techniques 2001 (Proceedings of Eurographics Workshop on Rendering)*. Springer, 2001.

[85] Peter Schröder, Wim Sweldens, Michael Cohen, Tony DeRose, and David Salesin. Wavelets in Computer Graphics. In *ACM SIGGRAPH 96 Course Notes*. 1996.

[86] Peter Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. In *Proceedings of Graphics Interface 90*, pages 205–212, 1990.

[87] Peter Shirley. Time Complexity of Monte Carlo Radiosity. *Computers and Graphics*, 16(1), 1992.

[88] Peter Shirley, Michael Ashikhmin, Michael Gleicher, Stephen R. Marschner, Eric Reinhard, Kelvin Sung, William B. Thompson, and Peter Willemsen. *Fundamentals of Computer Graphics, Second Edition*. AK Peters, 2005.

[89] François X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A Global Illumination Solution for General ReflectanceDistributions. *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, 25(4):187–196, 1991.

[90] François X. Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, 1994.

[91] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, 2003.

[92] 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 (Proceedings of ACM SIGGRAPH 2002)*, pages 527–536, 2002.

[93] Peter-Pike Sloan, Xinguo Liu, Heung-Yeung Shum, and John Snyder. Bi-scale radiance transfer. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, 2003.

[94] Brian Smits. *Efficient hierarchical radiosity in complex environments*. PhD thesis, Cornell University, 1994.

[95] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. *Computer Graphics (Proceedings of ACM SIGGRAPH 94)*, 28:435–442, 1994.

[96] Brian Smits, James Arvo, and David Salesin. An importance-driven radiosity algorithm. *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, pages 273–282, 1992.

[97] Marc Stamminger, Hartmut Schirmacher, Philipp Slusallek, and Hans-Peter Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Forum (Proceedings of Eurographics 98)*, 17(3), 1998.

[98] A. James Stewart and Sherif Ghali. Fast computation of shadow boundaries using spatial coherence and backprojections. *Computer Graphics (Proceedings of ACM SIGGRAPH 94)*, pages 231–238, 1994.

[99] Eric J. Stollnitz, Anthony D. DeRose, and David H. Salesin. *Wavelets for Computer Graphics (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers, Inc., 1996.

[100] Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic Computer Graphics: Modeling, Rendering and Animation*. Morgan Kaufmann Publishers Inc, 2002.

[101] Wolfgang Stürzlinger. Adaptive mesh refinement with discontinuities for the radiosity method. In *Proceedings of Eurographics Workshop on Rendering 94*, pages 239–248, 1994.

[102] Frank Suykens and Yves D. Willems. Adaptive filtering for progressive monte carlo image rendering. In *Proceedings of WSCG 2000*, 2000.

[103] Jörg Schmittler Sven Woop and Philipp Slusallek. Rpu: A programmable ray processing unit for realtime ray tracing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, 2005.

[104] Eric Tabellion and Arnauld Lamourlette. An approximate global illumination system for computer generated films. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004)*, 23(3):469–476, 2004.

[105] Seth Teller and Pat Hanrahan. Global visibility algorithms for illumination computations. In *Proceedings of ACM SIGGRAPH 93*, pages 239–246. ACM Press, 1993.

[106] Patrick C. Teo, Eero P. Simoncelli, and David J. Heeger. Efficient linear re-rendering for interactive lighting design. Technical Report STAN-CS-TN-97-60, Stanford University, 1997.

[107] Roy Troutman and Nelson L. Max. Radiosity algorithms using higher order finite element methods. In *Proceedings of ACM SIGGRAPH 93*, pages 209–212, 1993.

[108] Eric Veach and Leonidas J. Guibas. Bidirectional estimators for light transport. In *Rendering Techniques 94 (Proceedings of Eurographics Workshop on Rendering)*, pages 147–162, 1994.

[109] Eric Veach and Leonidas J. Guibas. Metropolis light transport. *Computer Graphics (Proceedings of ACM SIGGRAPH 97)*, 31:65–76, 1997.

[110] Carsten Wachter and Alexander Keller. Instant ray tracing: The bounding interval hierarchy. In *Rendering Techniques 2006 (Eurographics Symposium on Rendering*, 2006.

[111] Ingo Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004.

[112] Gregory J. Ward and Paul Heckbert. Irradiance Gradients. In *Proceedings of Eurographics Workshop on Rendering*, pages 85–98, 1992.

[113] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics (Proceedings of ACM SIGGRAPH 88)*, 22:85–92, 1988.

[114] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.

[115] Lance Williams. Casting curved shadows on curved surfaces. *Computer Graphics (Proceedings of ACM SIGGRAPH 78)*, pages 270–274, 1978.

[116] Andrew J. Willmott. *Hierarchical Radiosity with Multiresolution Meshes*. PhD thesis, Carnegie Mellon University, 2000.

[117] Andrew Woo, Pierre Poulin, and Alain Fournier. A survey of shadow algorithms. *IEEE Computer Graphics and Applications*, 10(6):13–32, 1990.

[118] Adam Worrall. *Dynamic Discontinuity Meshing*. PhD thesis, University of Bristol, 1998.

[119] Harold R. Zatz. Galerkin radiosity: a higher order solution method for global illumination. In *Proceedings of ACM SIGGRAPH 93*, pages 213–220. ACM Press, 1993.

[120] Kun Zhou, Yaohua Hu, Stephen Lin, Baining Guo, and Heung-Yeung Shum. Precomputed shadow fields for dynamic scenes. *ACM Transactions on Graphics (Proceedings ACM SIGGRAPH 2005)*, 24(3):1196–1201, 2005.

[121] Sergey Zhukov, Andrey Iones, and Grigorij Kronin. An ambient light illumination model. In *Rendering Techniques 98 (Proceedings of Eurographics Workshop on Rendering)*, pages 45–55, 1998.